UNIVERSIDADE FEDERAL DE SERGIPE

PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Aplicação de redes neurais recorrentes no reconhecimento automático da fala em ambientes com ruídos

Dissertação de Mestrado

Luciana Maiara Queiroz de Santana

São Cristóvão – Sergipe

2017

UNIVERSIDADE FEDERAL DE SERGIPE

PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Luciana Maiara Queiroz de Santana

Aplicação de redes neurais recorrentes no reconhecimento automático da fala em ambientes com ruídos

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de mestre em Ciência da Computação.

Orientador(a): Prof. Dr. Leonardo Nogueira Matos

São Cristóvão – Sergipe

2017

Luciana Maiara Queiroz de Santana

Aplicação de redes neurais recorrentes no reconhecimento automático da fala em ambientes com ruídos/ Luciana Maiara Queiroz de Santana. – São Cristóvão – Sergipe, 2017-

68 p. : il.

Orientador: Prof. Dr. Leonardo Nogueira Matos

Dissertação de Mestrado – UNIVERSIDADE FEDERAL DE SERGIPE
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO, 2017.

1. Reconhecimento automático da fala. 2. Redes Neurais Recorrentes. I. Prof. Dr. Leonardo Nogueira Matos. II. Universidade Federal de Sergipe. III. Aplicação de redes neurais recorrentes no reconhecimento de fala em ambientes com ruído

CDU 02:141:005.7

Luciana Maiara Queiroz de Santana

Aplicação de redes neurais recorrentes no reconhecimento automático da fala em ambientes com ruídos

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de mestre em Ciência da Computação.

Trabalho aprovado. São Cristóvão – Sergipe, 26 de Julho de 2017:

Prof. Dr. Leonardo Nogueira Matos
Orientador

Prof. Dr. Hendrik Teixeira Macedo
Professor interno

Prof. Dr. Paulo Salgado Gomes de Mattos Neto
Professor convidado

São Cristóvão – Sergipe
2017

UNIVERSIDADE FEDERAL DE SERGIPE
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
NÚCLEO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Relatório de defesa pública do (a) Senhor (a) **LUCIANA MAIARA QUEIROZ DE SANTANA** no Programa de Ciência da Computação (PROCC) da UFS.

Aos 26 dias de julho de 2017, realizou-se a **Defesa de Mestrado** do trabalho intitulado **"Aplicação de redes neurais recorrentes no reconhecimento automático da fala em ambientes com ruídos"** sob orientação do Profº. Drº. **Leonardo Nogueira Matos**

Depois de declarada aberta a sessão, o Presidente da Banca passou inicialmente a palavra ao candidato para exposição e a seguir aos examinadores para as devidas arguições que se desenvolveram nos termos regimentais. Em seguida, a comissão julgadora proclamou o resultado:

Nome Banca Examinadora	Instituição	Nota (de 0 a 10,0)	Assinatura
Leonardo Nogueira Matos	UFS		
Hendrik Teixeira Macedo	UFS		
Paulo Salgado Gomes de Mattos Neto	UFPE		
	Média =		

Dessa maneira o Resultado Final é: X APROVADO ou REPROVADO

Parecer da Banca Examinadora *

- Obs: Se o candidato for reprovado, o preenchimento do parecer é obrigatório.

São Cristóvão/SE

Assinatura do Orientador:

Luciana Maiara Q. de Sant

Assinatura do Aluno:

Agradecimentos

Primeiramente agradeço a Deus por ter me concedido saúde e força para conclusão deste trabalho.

Aos meus pais Fátima e Gildeon pelo amor, paciência e incentivo constante, sendo a base de todo meu crescimento pessoal, acadêmico e profissional.

Ao namorado Pedro, pela paciência, incentivo e apoio incondicional.

À minha vó Lúcia, minha irmã Allana, meus padrinhos Carla e Tinho, meu sobrinho Matheus, meus tios e todos os familiares por estarem sempre presentes em minha vida, torcendo por mim.

Ao orientador Prof. Dr. Leonardo Nogueira Matos pela confiança depositada em mim e por todo suporte necessário para a realização deste estudo.

Aos demais professores e funcionários da Universidade Federal de Sergipe.

À CAPES, pela concessão da bolsa de mestrado

“A tua palava é lâmpada que ilumina os meus passos
e luz que clareia o meu caminho.
(Bíblia Sagrada, Salmos 119, 105)

Resumo

Inúmeras tarefas de aprendizagem exigem lidar com dados sequenciais, a exemplo de tradutores de textos, geradores de músicas, entre outros. Os sistemas que utilizam redes neurais profundas têm mostrado resultados promissores no reconhecimento automático de fala, onde um dos maiores desafios é o reconhecimento em sinais de voz contaminados com ruído. Para este trabalho, combinamos duas arquiteturas conhecidas de aprendizagem profunda, as redes neurais convolucionais para abordagem acústica e uma arquitetura recorrente com classificação temporal conexionista para modelagem sequencial. As redes neurais recorrentes são modelos que capturam a dinâmica da sequência através de uma topologia que contém ciclos, ao contrário das redes neurais acíclicas ou de alimentação direta (*feedforward*). O modelo estudado neste trabalho é um caso particular de rede recorrente profunda que, ao contrário de seus correlatos de arquitetura rasa, é capaz de reter um estado que pode representar informações de uma janela de contexto arbitrariamente longa. Os resultados experimentais mostraram que a arquitetura proposta alcançou um desempenho superior quando comparado ao modelo clássico, modelo oculto de Markov, em testes realizados sobre as mesmas bases de dados.

Palavras-chave: Reconhecimento Automático de Voz, Ruído Aditivo, Aprendizado Profundo, Rede Neural Recorrente.

Abstract

Many learning tasks require dealing with sequential data, such as text translators, music generators, and more. Deep Neural Networks have shown promising results in automatic speech recognition, where one of the main challenges is voice recognition signals in the presence of noise. In this manuscript, we combine two known deep learning architectures, Convolutional Neural Networks for acoustic modeling, and a recurrent architecture with Classification Temporal Conexionist for sequential modeling. Recurrent Neural Networks (RNN) are models that capture sequence dynamics through a topology that contains cycles, unlike acyclic neural networks or feedforward networks. The RNN studied in this work is a particular case of a deep learning network that, unlike its shallow correlates, it is able to retain a state that can represent information from an arbitrarily long context window. The experimental results showed that the proposed architecture achieved superior performance when compared to Hidden Markov Model in tests carried out on the same databases.

Keywords: Automatic Speech Recognition, Additive Noise, Deep Learning, Recurrent Neural Network.

Lista de ilustrações

Lista de tabelas

Lista de Algoritmos

Lista de abreviaturas e siglas

ANN	Redes Neurais Artificiais
ASR	Reconhecimento Automático de Fala
BLSTM	Memória de Longo Curto Termo Bidirecional
CNN	Convolutional Neural Network
CTC	Connectionist Temporal Classification
DNN	Deep Neural Network
GMM	Gaussian Mixture Models
HMM	Cadeia Oculta de *Markov*
HZ	*Hertz*
LPC	Coeficientes de Predição Linear
LSTM	Memória de Longo Curto Termo
LM	Modelo Linguístico
MFCC	Coeficientes mel-*cepstrais*
MLP	Perceptron Multi-Camadas
MSE	Erro Quadrático Médio
RNN	Rede Neural Recorrente
SNR	Relação sinal-ruído
SVM	Máquina de vetores de suporte

Sumário

1 Introdução

A fala é um sinal complexo variável no tempo com correlações complexas em diferentes escalas de tempo, sendo produzida de maneira diferente por cada pessoa (LIPTON; BERKOWITZ; ELKAN, 2015). A linguagem falada é a forma mais natural de comunicação humana, todavia realizar este processo computacionalmente é uma tarefa bastante difícil, podendo ser influenciada por uma série de fatores, limitando assim, o desempenho desses sistemas. Podemos citar como exemplo:

- A variabilidade do transdutor e do canal, como microfones, telefones fixos e celulares;
- A variabilidade do ruído de fundo gerado a partir de outras vozes, carros, ar-condicionado, dentre outros;
- A variabilidade intra-locutor resultante de mudanças do estado físico/emocional dos locutores, velocidade de pronúncia ou qualidade da voz, movimentos da boca, ruídos da respiração, hesitação ao falar, etc.;
- A variabilidade entre-locutores originado pelas diferenças na condição sociocultural, tamanho do trato vocal e dialeto de cada pessoa;

O reconhecimento automático da fala, do inglês *Automatic Speech Recognition* (ASR), tem sido utilizado cada vez mais pelo homem, através de celulares, tablets e aplicativos que facilitam a iteração homem *versus* máquina, como pesquisa por voz e assistentes virtuais de fala (por exemplo, Siri da *Apple, Google Now* do *Android* e Cortana da *Microsoft*). Os sistemas ASR podem ser caracterizados de maneiras diferentes, a saber: o modo de pronúncia, tamanho do vocabulário, entre outros (FURUI, 2000). O modo da pronúncia refere-se ao reconhecimento de palavras isoladas ou de fala contínua. O tamanho do vocabulário está relacionado à quantidade de palavras a ser reconhecida.

As abordagens dominantes até os anos 2010 baseavam-se em Modelos Ocultos de *Markov* (*Hidden Markov Model*, HMM) para modelar a estrutura sequencial da fala, com cada estado HMM usando um modelo de Mistura de Gaussianas (*Gaussian Mixture Model*, GMM) para modelar uma representação espectral da onda sonora (MOHAMED; DAHL; HINTON, 2012). Porém, essa abordagem é sensível ao ruído introduzido pelo ambiente (SELTZER; YU; WANG, 2013).

Paralelamente, na última década as pesquisas em redes neurais foram retomadas com maior intensidade quando Redes Neurais Profundas (*Deep Neural Network*, DNN), venceram concursos internacionais oficiais de reconhecimento de padrões no ICDAR (*International Conference on Document Analysis and Recognition*) em 2009, obtendo os primeiros resultados no reconhecimento de padrões visuais. Foram três competições de reconhecimento de manuscritos em três línguas diferentes (Francês, árabe e Farsi). (SCHMIDHUBER, 2015).

Ao contrário das redes neurais rasas, do inglês *shallow learning*, as redes neurais profundas referem-se a uma classe de técnicas de aprendizagem de máquina onde muitas camadas de processamento de informações são exploradas, possuindo assim um melhor potencial para extrair representações dos dados brutos para criar modelos mais complexos. DNN fornece uma modelagem temporal, mas limitada apenas por operações em tamanhos fixos, sendo assim inadequadas para lidar com dependências de longo prazo (SAK; SENIOR; BEAUFAYS, 2014). A Rede Neural Recorrente, do inglês *Recurrent Neural Network* (RNN), por sua vez, é uma rede profunda adequada para tarefas de modelagem sequencial onde sua arquitetura contêm ciclos que alimentam as ativações da rede a partir de entradas de tempos anteriores influenciando previsões no tempo corrente. Estas ativações são armazenadas nos estados internos da rede, que podem em princípio, manter informações contextuais temporais de longo prazo. Entre alguns modelos de RNN, existem a Rede de Jordan (JORDAN, 1986), Rede Elman (ELMAN, 1990), e a Memória de Longo Curto Prazo (LSTM, *Long Short-Term Memory*) (HOCHREITER; SCHMIDHUBER, 1997).

Em particular, LSTM possui uma arquitetura capaz de armazenar informações em células de memórias por um período mais longo de tempo e com capacidade de aprender grande quantidade de informações relevantes para a tarefa de regressão ou classificação (WÖLLMER et al., 2013). Correlato à rede LSTM, Graves e Schmidhuber (2005) propuseram a rede Bidirecional LSTM (BLSTM) capaz de superar a limitação de uma única direção, consistindo de um processamento em duas direções, para a frente e para trás no tempo.

Neste trabalho exploramos o uso das máquinas de reconhecimento de dados sequenciais HMM, LSTM e BLSTM com o objetivo de analisar a robustez do sistema quando o sinal de áudio é contaminado com ruído aditivo. Embora este não seja o primeiro trabalho a empregar estas máquinas de aprendizado profundo no contexto de reconhecimento da fala, mostramos que a abordagem proposta mostrou-se adequada no reconhecimento de palavras isoladas na presença de ruído, apresentando desempenho superior a outros trabalhos que exploraram a mesma base de

dados como em Santos et al. (2015) e Raulino, Duarte e Montalvao (2014).

O reconhecimento de fala de palavras isoladas pode potencialmente ser utilizado em uma ampla variedade de aplicações, como nas que empregam acionamento de comandos através de interface de voz. Os dispositivos móveis, por apresentar limitações de memória e processamento, bem como por possuir restrições de tamanho, o que naturalmente impõe limitações no teclado, são dispositivos que podem fazer melhor uso deste tipo de aplicação. Por esta razão acreditamos que o estudo apresentado nesta dissertação pode ter desdobramentos práticos ou ajudar a avançar a área de desenvolvimento de aplicações para estes dispositivos.

1.1 Histórico

Sistemas ASR têm sido foco de estudo já há algumas décadas. O primeiro ocorreu na década de 1950, nos laboratórios BELL, através da criação de um reconhecedor de dígitos isolados apenas por um único locutor (ANUSUYA; KATTI, 2010).

Na década de 60, foram desenvolvidos métodos de comparação de padrões de sequências, por meio de uma abordagem determinística chamada Alinhamento Temporal Dinâmico, do inglês *Dynamic Time Warp* (DTW) proposto por Vintsyuk (1968) e uma abordagem estatística chamada Modelo Escondido de Markov, do inglês *Hidden Markov Models* (HMM) proposto por Baum e Petrie (1966), nessa década, eram necessários 50 computadores para efetuar o reconhecimento de dígitos (MOREIRA et al., 2012).

Na década de 70, a Agência de Projetos de Investigação Avançados (*Advanced Research Projects Agency*, ARPA) subsidiou um vasto projeto de compreensão da fala. O objetivo era efetuar o reconhecimento automático da fala com um vocabulário de 1000 palavras, usando um pequeno número de locutores, fala contínua e uma gramática restrita com menos de 10% de erro semântico (HARRIS, 1985). O sistema foi desenvolvido por um estudante da Universidade de Carnegie Mellon, chamado Bruce Lowerre. Os fundos de investimento foram de 15 milhões de dólares.

Na década de 80, HMM era a ferramenta predominante no reconhecimento de fala. As redes neurais também foram reintroduzidas no final dos anos 80 aplicadas a problemas de reconhecimento de fala (ANUSUYA; KATTI, 2010). A utilização dessas tecnologias foram um grande incentivo para a implementação de sistemas robustos de reconhecimento de fala contínua em grandes vocabulários, como por exemplo o sistema BYBLOS (KUBALA et al., 1988) e SPHINX (LEE, 1988)

Em 1986 foi criado a primeira grande base de dados a ser usada por grande parte da comunidade científica, conhecida como TIMIT[1]. Foi escolhido um alfabeto de 61 fonemas para representar todas as distinções fonéticas. Foram 630 locutores que proferiram 10 frases cada,

[1] https://catalog.ldc.upenn.edu/ldc93s1

sendo que duas delas são comuns a todos. Esta constitui, ainda hoje, uma das maiores e mais utilizadas bases de dados existentes. Os dados foram gravados e segmentados foneticamente no Instituto de Tecnologia de *Massachussetts* (MIT) nos Estados Unidos. Ainda nesse ano, surgiu a rede neural recorrente de Jordan (JORDAN, 1986), em que no seu treinamento não utilizava a retropropagação do erro, treinando assim de forma superficial, nesse caso, apenas era possível ver um passo pra trás no tempo (SCHMIDHUBER, 2015).

Na década de 1990, a utilização do *backpropagation* foi disseminada para alguns tipos de redes neurais, como o caso da máquina CNN (*Neocognitron*) discutida inicialmente por Fukushima (1981), porém, só no trabalho descrito em LeCun et al. (1990) o *backpropagation* foi aplicado. Este trabalho também introduziu o conjunto de dados MNIST de dígitos manuscritos, que ao longo do tempo tornou-se talvez o *benchmark* mais famoso da Aprendizagem de Máquinas. A máquina CNN ajudou a alcançar um bom desempenho no MNIST (LECUN et al., 1990) e no reconhecimento de impressões digitais (BALDI; CHAUVIN, 1993).

Na década 2000, um marco importante foi a popularização do aprendizado profundo (GOODFELLOW; BENGIO; COURVILLE, 2016). Estes modelos usam um princípio mais geral de aprender em múltiplos níveis de composição. A arquitetura mais complexa e a existência de novos algoritmos para ajuste de pesos e parâmetros, os diferenciam de seus correlatos mais simples, chamados redes rasas, do inglês *shallow learning* (BENGIO, 2013). Segundo Goodfellow, Bengio e Courville (2016), aprendizado profundo remonta à década de 1940, onde utilizavam outros nomes, como: *cybernetics* entre os anos de 1940 e 1960, e *connectionism* nos anos de 1980 e 1990, e o ressurgimento atual no início do aprendizado nomeado aprendizado de máquina em 2006.

Desde 2006, o aprendizado profundo emergiu como uma nova área de pesquisa em aprendizagem de máquinas (HINTON; OSINDERO; TEH, 2006). Nos últimos anos, para Deng, Yu et al. (2014), as técnicas desenvolvidas a partir de pesquisas de aprendizado profundo já impactaram uma ampla gama de trabalho de processamento de sinal e informação dentro dos novos e ampliados escopos, incluindo o reconhecimento automático da fala (HINTON et al., 2012).

1.2 Objetivos

O objetivo principal deste trabalho é avaliar o impacto do uso das redes recorrentes no reconhecimento da fala em ambientes com ruídos.

Objetivos Específicos

- Apresentar e avaliar uma abordagem para reconhecimento de fala utilizando a rede profunda LSTM.
- Investigar diferentes máquinas sequenciais para o reconhecimento automático da fala.

- Fazer uma análise comparativa com outros métodos propostos na literatura.
- Propor uma rotulação automática a nível acústico do sinal.

1.3 Problemática e Hipótese

Para a tarefa de classificação, são utilizados algoritmos de aprendizagem de máquina supervisionado, cujo objetivo é ser capaz de produzir uma boa generalização a partir dos dados empregados no treinamento. Isto é, procura-se obter máquinas que produzam uma baixa taxa de erro durante a classificação, quando lhe são apresentados elementos não vistos. Este erro, segundo uma análise realizada em Geman, Bienenstock e Doursat (1992), pode ser decomposto através da soma de duas partes conflitantes: a variância e o quadrado do viés. Os autores afirmam que, à medida que se aumenta a complexidade do modelo, isto é, à medida em que a máquina se torna mais complexa, o quadrado do viés tem seu valor diminuído, enquanto que a variância tem seu valor aumentado, sendo o ideal, segundo os autores, encontrar um compromisso entre um modelo complexo (alta variância e baixo viés) e generalizável (baixa variância e alto viés) (Figura 1).

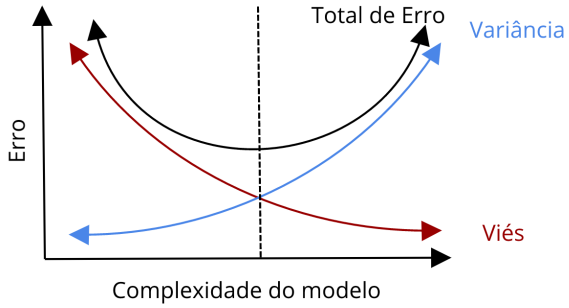

Figura 1 – Representação gráfica do teorema do viés-variância

Esse dilema pode ser exemplificado através de um problema de regressão apresentado na Figura 2, onde a linha vermelha representa o padrão esperado, os pontos azuis representam as amostras para treinamento e o objetivo é aplicar uma função que minimize o erro encontrado na busca pelos padrões dos dados a partir das amostras para treinamento. Para tal, foram utilizados funções polinomiais de diferentes graus, representando diferentes níveis de complexidade, a saber: polinômio de grau 2 (Figura 3), grau 6 (Figura 4) e grau 15 (Figura 5).

O exemplo da Figura 3 utilizando o polinômio de grau 2 (baixa complexidade) apresenta um baixa variância com valor 1.5 e alto viés com valor de $\sqrt{16.1}$. O valor do erro médio encontrado (MSE - *Mean Square Error*) é 17.6. Com um aumento na complexidade do modelo polinômio de grau 6, exibido na Figura 4, o valor da variância aumenta para 2.5 enquanto que o viés diminui para $\sqrt{5.4}$, consequentemente o valor de MSE também diminui, chegando a 8.9. No

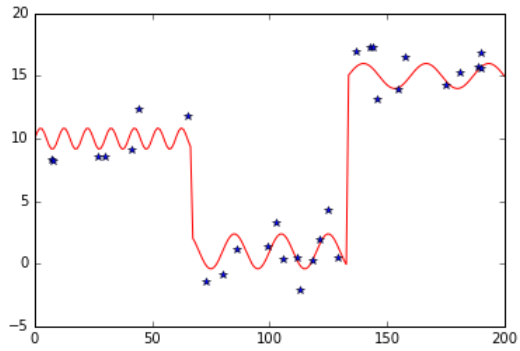

Figura 2 – Exemplo utilizando regressão

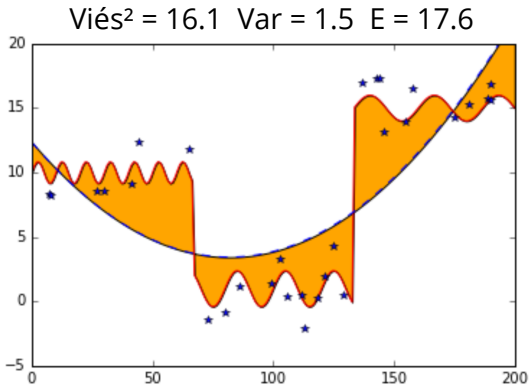

Figura 3 – Função polinomial de grau 2

exemplo da Figura 5, quando é utilizado um polinômio de grau 15, o viés continua a diminuir enquanto que a variância e o erro total aumentam consideravelmente.

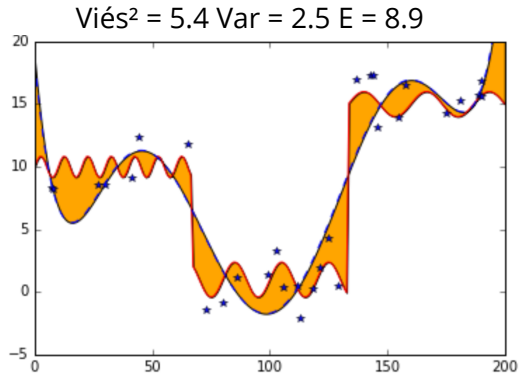

Figura 4 – Função polinomial de grau 6

Este exemplo ilustra o fato de que o erro total diminui à medida que o grau do polinômio aumenta, depois cresce bastante, tal como previsto teoricamente. Então, para obter um bom compromisso entre generalização e super-especialização em casos nos quais o padrão é localmente

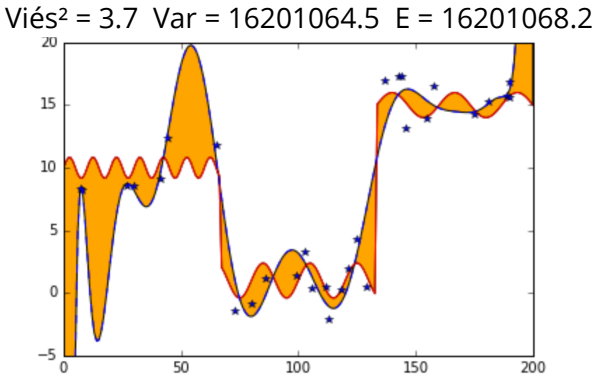

Figura 5 – Função polinomial de grau 15

estacionário, pode-se empregar uma composição de máquinas localmente especializadas e um mecanismo para tratar a predição individual de cada uma delas. Neste exemplo, a composição de máquinas está ilustrada na Figura 6 e é a solução que, em média, produz o menor erro quadrático.

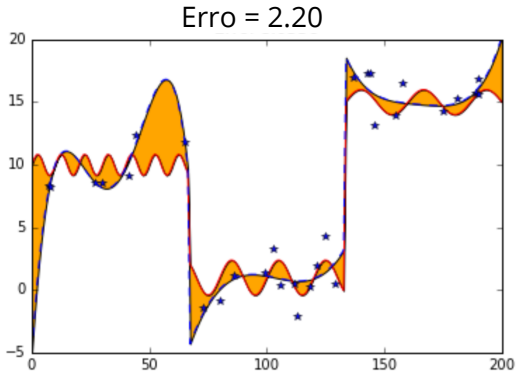

Figura 6 – Funções polinomiais de grau 6 aplicadas a cada padrão estacionário

No problema abordado neste trabalho, admitimos por hipótese que os padrões da fala em curtos segmentos de tempo são estacionários, ou seja, suas propriedades estatísticas são invariantes ainda que este seja um padrão aleatório. Embora não tenhamos analisado a estacionariedade do sinal, na literatura, alguns autores como Song e Cai (2015), Abdel-Hamid et al. (2014) e Jaitly (2014) utilizaram janelas de tempo com largura de 15 a 30 ms, pois admitiam existir preservação da estacionariedade do sinal nestes segmentos. No nosso trabalho propomos utilizar uma máquina para classificar elocuções fonéticas, padrões que supostamente são estacionários, e uma máquina sequencial para modelar a estrutura de encadeamento desses padrões. É desta forma que buscamos encontrar o balanço entre viés e variância, tendo em vista que o sistema como um todo é resultado da composição de duas máquinas, cada uma especializada em uma parte da solução.

Esta hipótese não é nova. De fato, como dito anteriormente, ela já foi experimentada

em outros trabalhos e, até onde estendeu nossa cobertura da literatura, é a metodologia usual para tratar o problema de reconhecimento de fala. O fato novo, que diferencia nosso trabalho, é o emprego de máquinas de aprendizado profundo, que normalmente exigem uma grande quantidade de amostras para treinamento, já que são inerentemente complexas, na composição de um modelo treinado com poucas amostras.

Procuramos mostrar através de experimentos que um modelo elaborado pela composição de redes neurais CNN e LSTM, ambas máquinas de aprendizado profundo, pode ser bem sucedido para classificação de palavras isoladas, ainda que treinado com bases pequenas, e ainda que, as máquinas utilizadas na composição não sejam bem sucedidas na solução de problema quando empregadas isoladamente. Nos nossos experimentos as bases empregadas possuem quantidade de amostras compatíveis com a que Kolář, Hradiš e Zemčík (2016) rotularam como "pequeno", isto é, possuem até 20 classes de rótulos e até 1000 amostras para compor a base de dados.

1.4 Trabalhos relacionados

Recentemente, a empresa Google anunciou melhorias para transcrição do Google Voice usando LSTM. Anteriormente, utilizavam o modelo GMM-HMM, o estado da arte no reconhecimento de fala há mais de 30 anos. Abordagens bem-sucedidas utilizando LSTM podem ser vistas também em Graves (2012a) e Eyben et al. (2009), que propõem uma utilização de uma arquitetura recorrente, que combina a Classificação Temporal Conexionista, do inglês *Connectionist Temporal Classification* (CTC) com uma rede recorrente, que prevê cada fonema de acordo com os dados anteriores, obtendo assim um modelo acústico e linguístico de forma conjunta (GRAVES; MOHAMED; HINTON, 2013). Os autores também demonstraram por meio de experimentos que uma rede profunda LSTM treinada com a regularização adequada consegue alcançar um erro de apenas 17.7% na base TIMIT, um *benchmark* de reconhecimento de fonemas. Esta parece ser a melhor pontuação registrada.

O emprego das Redes Neurais Profundas, do inglês, *Deep Neural Network* (DNN) em tarefas de reconhecimento de fala na presença de ruído foi explorado em alguns trabalhos, a exemplo de Seltzer, Yu e Wang (2013), Narayanan e Wang (2014) e Li et al. (2014). Em Santos et al. (2015) foi proposto um modelo híbrido composto por Redes Neurais Convolucionais, do inglês *Convolutional Neural Network* (CNN) e HMM (Figura 7), onde os autores sugerem que os *frames* do sinal da fala no domínio da frequência possam ser usados como entrada para CNN, que por sua vez gera um vetor com a classificação fonética da elocução, usando subsequentemente uma máquina de reconhecimento de dados sequenciais - HMM. Nessa abordagem, a rede CNN possui o papel de modelar uma representação acústica dos *frames* da fala, enquanto que o modelo HMM modela a estrutura temporal e as dependências entre *frames* adjacentes.

O principal trabalho relacionado se deve a Santos et al. (2015). Esta pesquisa é na

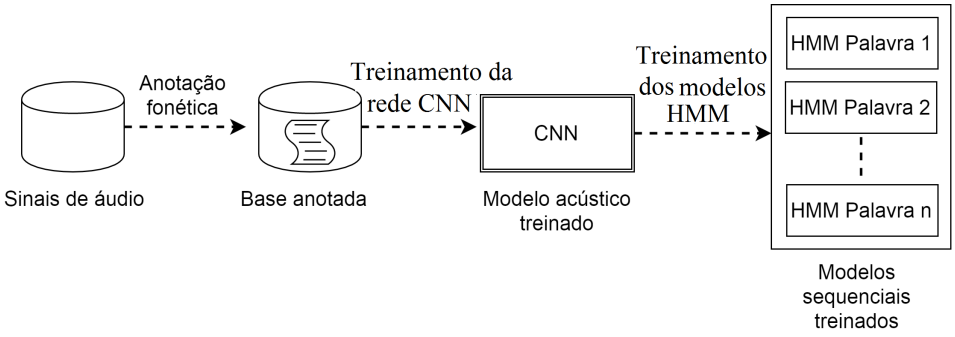

Figura 7 – Abordagem híbrida utilizando CNN e HMM proposta em Santos et al. (2015)

realidade uma extensão desta primeira. Em Santos et al. (2015) foi comparado diferentes máquinas de aprendizagem para a modelagem acústica, tais como o modelo de Mistura de Gaussianas, do inglês *Gaussian Mixture Model* (GMM), Máquina de Vetores de Suporte do inglês *Support Vector Machine* (SVM) e CNN, onde percebeu-se que CNN foi a menos influenciada por ruídos. Acredita-se que a sua robustez é influenciada pelo compartilhamento de pesos e conectividade local que introduz um certo nível de invariância a distorções presentes em ambientes não controlados. Baseado nisso, foi escolhido a CNN para realizar a modelagem acústica neste trabalho. Além disto, o presente trabalho também realizou comparações entre diferentes tipos de arquiteturas sequenciais, como o LSTM, BLSTM e HMM.

1.5 Organização da dissertação

Os próximos capítulos estão organizadas da seguinte maneira. O capítulo II traz uma breve revisão da literatura e o detalhamento das abordagens utilizadas. No capítulo III, apresentamos as bases de dados e a metodologia aplicada nos experimentos. No capítulo IV, os resultados dos experimentos são apresentados e discutidos. Na capítulo V, é apresentado a conclusão e as sugestões para trabalhos futuros.

2

Fundamentação teórica

Neste capítulo será apresentado o referencial teórico do reconhecimento da fala, a modelagem acústica do sinal de voz e modelagem sequencial.

2.1 Reconhecimento Automático da Fala

O funcionamento de sistemas ASR (Figura 8) é dado primeiramente pela conversão de um sinal acústico produzido pelo homem em um sinal digital de áudio, podendo ser aplicadas algumas técnicas para eliminar os ruídos, períodos de silêncio, etc. Em seguida, é realizada a extração de características, conhecida também como *front-end*, onde o sinal acústico é representado em um espaço de dimensão menor, preservando as informações importantes, ou seja, evidenciando características do sinal que contribuem para sua identificação. As características tipicamente podem ser obtidas por métodos de bancos de filtros (*Filter Bank*), coeficientes mel-cepstrais (*Mel-frequency Cepstral Coefficients*, MFCC) (DAVIS; MERMELSTEIN, 1980) e coeficientes de predição linear (*Linear Predictive Coding*, LPC) (ICHIKAWA; NAKANO; NAKATA, 1973).

Neste trabalho, assim como em Santos et al. (2015), foram extraídas pelo método banco de filtros na escala mel (STEVENS; VOLKMANN; NEWMAN, 1937), sendo este considerado um método eficiente para representar as características do sinal da fala que são importantes para a informação do trato vocal. Essas características são processadas pelo modelo acústico, um modelo estatístico utilizado para calcular a verossimilhança da geração de observações acústicas a nível de fonemas (JAITLY, 2014). Este modelo refere-se ao processo de definir representações estatísticas para as sequências de vetores de características de cada um dos sons distintos que compõe uma palavra. Cada representação definida recebe um rótulo chamado fonema.

O decodificador é responsável pelo processo de busca no qual uma sequência de vetores correspondentes a características acústicas do sinal de voz é comparada com modelos de palavras.

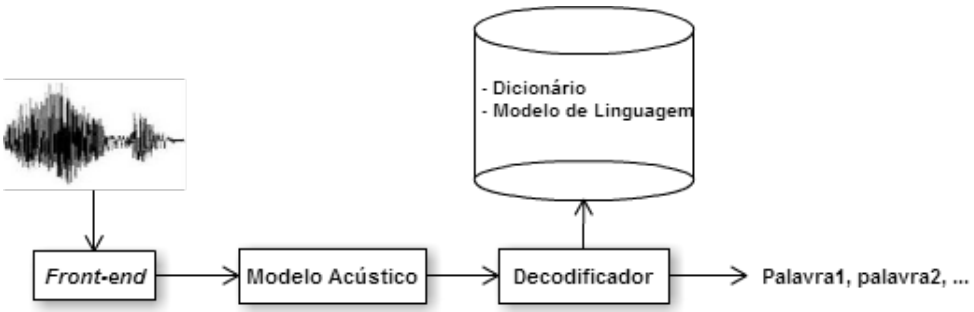

Figura 8 – Diagrama em blocos de um sistema de reconhecimento de fala genérico. Adaptado de Jaitly (2014)

Esta etapa é composta pelo dicionário e modelo linguístico. O dicionário (*pronunciation dictionary*) mapeia sequências de fonemas em palavras (JAITLY, 2014). Frequentemente uma palavra pode ter várias pronúncias alternativas, sendo necessário que o dicionário tenha várias entradas para uma mesma palavra. O modelo linguístico (LM, *language model*) estima a probabilidade de uma sequência de palavras de uma mesma sentença, em função das palavras que a antecedem. O modelo mais comum é o *N-gram* (MOHRI; PEREIRA; RILEY, 2002). No entanto, modelos linguísticos avançados tal como, RNNLM (*Recurrent Neural Network Language Models*) podem ser usados em cascatas de *n-gram*, fornecendo uma maneira rica e poderosa para modelar dados sequenciais (MIKOLOV et al., 2011). Williams et al. (2015) afirmam que as mais recentes evoluções dos sistemas ASR baseiam-se em RNNLM.

2.2 Rede Neural Convolucional

As redes neurais convolucionais, do inglês, *Convolutional Neural Network* (CNN), são um tipo de Redes Neurais Artificiais (ANN, Neural Networks Artificial) *feedfoward* inspirada no córtex visual dos animais (HUBEL; WIESEL, 1968), porém sua arquitetura garante que sejam muito mais robustas que a MLP (*Multi-Layer Perceptron*) em relação ao deslocamento, escala e distorção dos dados de entrada através dos campos receptivos locais, parâmetros compartilhados e amostragem (LECUN et al., 1998). Sua arquitetura é apresentada através da Figura 9, onde é possível observar que a rede pode ser formada por uma ou mais camadas de convolução e subamostragem.

A camada convolucional é composta por um conjunto de filtros também conhecidos como *kernel* de convolução. Cada filtro é aplicado à toda imagem de entrada resultando em um mapa de característica (*feature map*). Para o caso da voz, o sinal de áudio, que é unidimensional, é transformado para uma representação 2D – o espectrograma – e analisado como uma imagem. No espectrograma, o eixo horizontal é o tempo, tal como no sinal original, enquanto que o eixo vertical é o eixo das frequências, assim pode-se verificar em quais bandas a energia se distribui ao longo do tempo. Uma propriedade importante nessa etapa é o compartilhamento de pesos

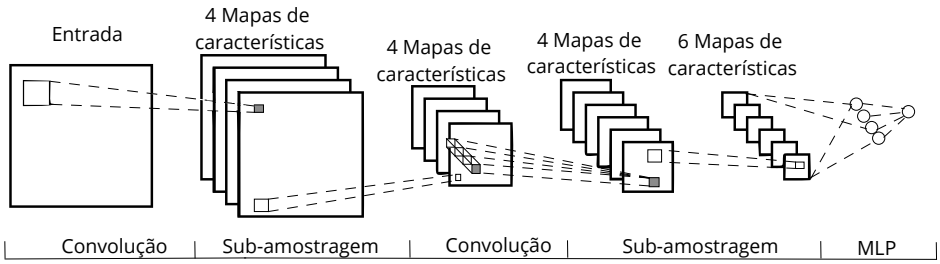

Figura 9 – Arquitetura genérica da CNN contendo 2 camadas de convolução e 2 de sub-amostragem e por fim uma MLP para calcular a saída da rede. Adaptado de LeCun et al. (1998)

entre os *kernels* do mesmo mapa de característica, como ilustra a Figura 10. Através disso, é possível detectar padrões independentes de sua localização na imagem de entrada.

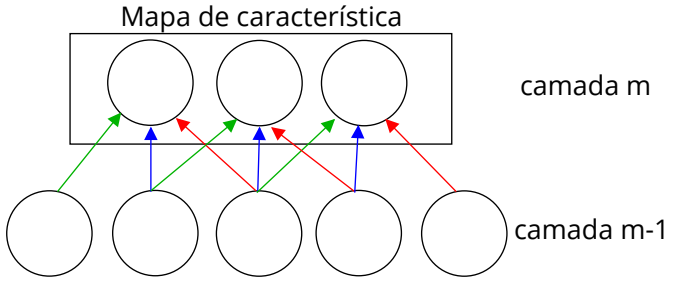

Figura 10 – Compartilhamento dos parâmetros para criação de um mapa de características. Adaptado de LeCun, Bengio e Hinton (2015)

Após a convolução, as ativações são passadas para uma segunda camada, a de subamostragem (*subsampling*), onde é possível calcular o valor máximo e médio de uma área de tamanho pré definido, gerando uma representação de entrada em resolução reduzida. O resultado é um outro mapa de características com resolução menor que proporciona invariância a pequenas translações (LECUN et al., 1998). Ao término dessas etapas uma MLP calcula a saída final da CNN. A saída é utilizada como entrada para o modelo sequencial, um modelo estatístico capaz de modelar uma sequência que forma uma palavra, através de um tipo de rede recorrente, chamada BLSTM.

2.3 Rede Neural Recorrente

As redes neurais recorrentes, do inglês *Recurrent Neural Network* (RNN) são usadas em tarefas que envolvem entradas sequenciais, como fala e linguagem. Essas redes possibilitam que as ligações entre neurônios formem ciclos, conhecidos como conexões recorrentes, que por sua vez, permitem que seu estado interno mantenha um "vetor de estado" que contém implicitamente informações sobre a história de todos os elementos passados da sequência. No tempo t, o nó com

a conexão recorrente recebe a entrada a partir dos dados correntes x^t e também dos valores do nó da camada oculta h^{t-1} do estado anterior da rede. A saída y^t no tempo t portanto é calculado dado os valores da camada oculta h^t, no tempo t e h^{t-1}, no tempo $t-1$.

A investigação sobre RNN teve início na década de 1980 (LIPTON; BERKOWITZ; ELKAN, 2015). No trabalho descrito por Jordan (1986) (Figura 11), o autor propôs uma arquitetura para aprendizagem supervisionada sobre sequências. Elman (1990) (Figura 12), descreveu uma arquitetura mais simples que a anterior. Em trabalhos subsequentes foram introduzidas a *Vanilla* (BENGIO; SIMARD; FRASCONI, 1994) e LSTM (HOCHREITER; SCHMIDHUBER, 1997). A diferença topológica entre elas está na ligação da recorrência. Na *Vanilla* (Figura 13), a recorrência é feita através de neurônios da mesma camada, a intermediária (ou oculta). Já a rede Elman, utiliza conexões da camada oculta para fazer a realimentação para uma unidade de contexto, uma camada com neurônios adicionais que representam uma unidade interna, e a rede Jordan utiliza os sinais da camada de saída para a unidade de contexto, ou seja, a conexão recorrente permite que a camada intermediária da rede acesse sua saída anterior, de modo que o comportamento subsequente possa ser modelado por respostas anteriores.

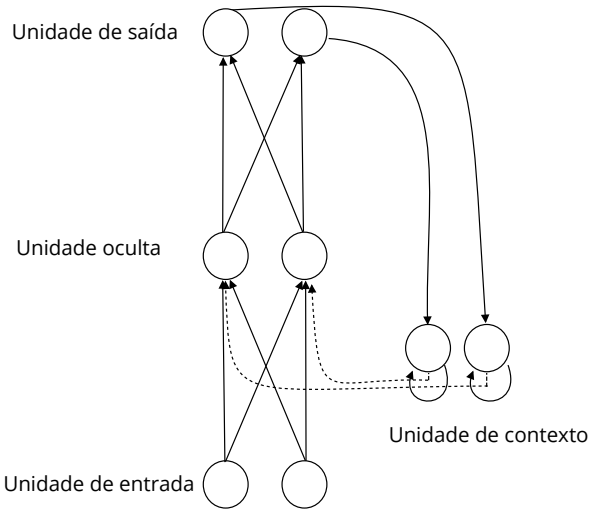

Figura 11 – Rede Jordan, adaptado de Jordan (1997)

O aprendizado com redes recorrentes tem sido considerado um desafio devido a dificuldade de aprender dependências de longo alcance como descrito em Bengio, Simard e Frasconi (1994). Os problemas chamados desaparecimento (*vanishing*) e explosão (*exploding*) do gradiente ocorrem durante a retropropagação do erro em redes com múltiplas camadas.

Introduzido em Bengio, Simard e Frasconi (1994), o problema da explosão do gradiente, refere-se ao grande aumento na norma do gradiente durante o treino. Tais eventos ocorrem devido à explosão dos componentes de longo prazo, que podem crescer exponencialmente. Já o desaparecimento do gradiente (Figura 14) refere-se ao comportamento oposto, quando a

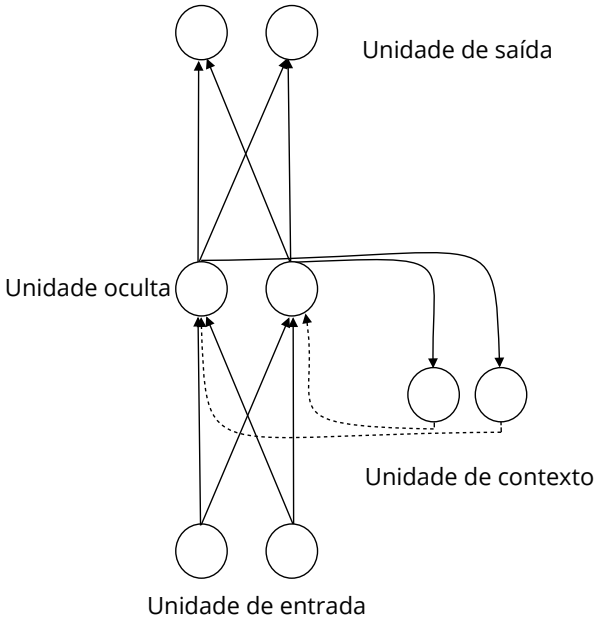

Figura 12 – Rede Elman, adaptado de Elman (1990)

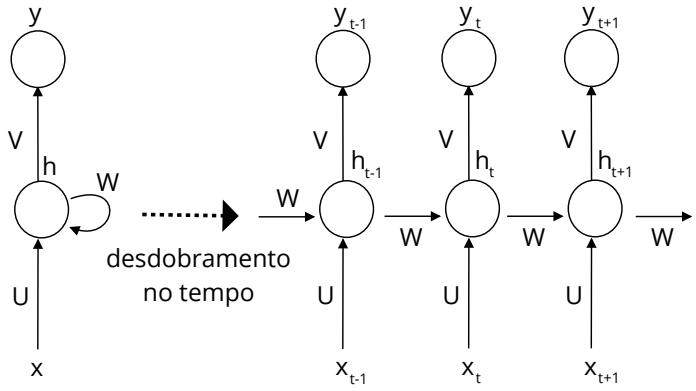

Figura 13 – *Vanilla* RNN, adaptado de LeCun, Bengio e Hinton (2015)

diferença na atualização dos pesos a longo prazo decaem rapidamente à norma 0, o que torna impossível para o modelo aprender a relação entre os eventos temporalmente distantes. Uma solução para este problema consiste na utilização de uma nova arquitetura RNN denominada *Long Short-Term Memory*, o qual possui uma arquitetura capaz de armazenar informações por um período mais longo de tempo e pode aprender grande quantidade de informações relevantes (WÖLLMER et al., 2013).

As redes recorrentes podem ser aplicadas a vários problemas do mundo real, tais como previsão da estrutura secundária de proteínas (HOCHREITER; SCHMIDHUBER, 1997), (BALDI et al., 2000) e (CHEN; CHAUDHARI, 2004), geração de música (ECK; SCHMIDHUBER, 2002), reconhecimento de fala (GRAVES, 2012a) (SONG; CAI, 2015), reconhecimento de escrita (GRAVES et al., 2008) e (LIWICKI; BUNKE, 2005) e geração de texto (SUTSKEVER;

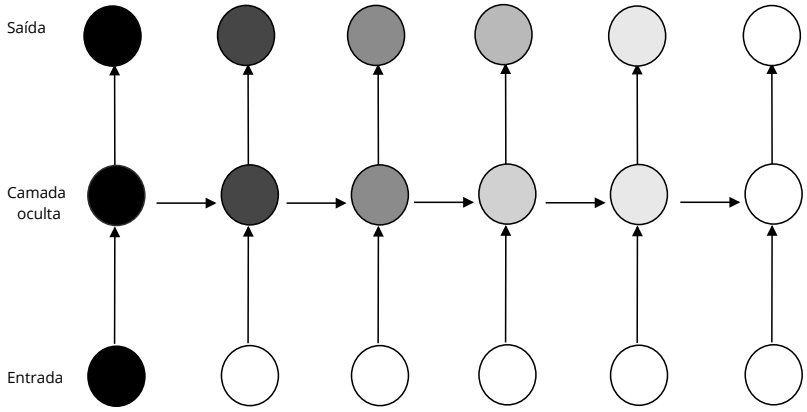

Figura 14 – Problema do desaparecimento do gradiente para RNN, adaptado de Graves (2012b)

MARTENS; HINTON, 2011).

2.3.1 Long Short-Term Memory

A rede de longo curto termo, do inglês *Long Short-Term Memory* (LSTM) (Figura 15), foi inicialmente proposta por Hochreiter e Schmidhuber (1997), para resolver o tratamento de sequências longas por uma rede recorrente.

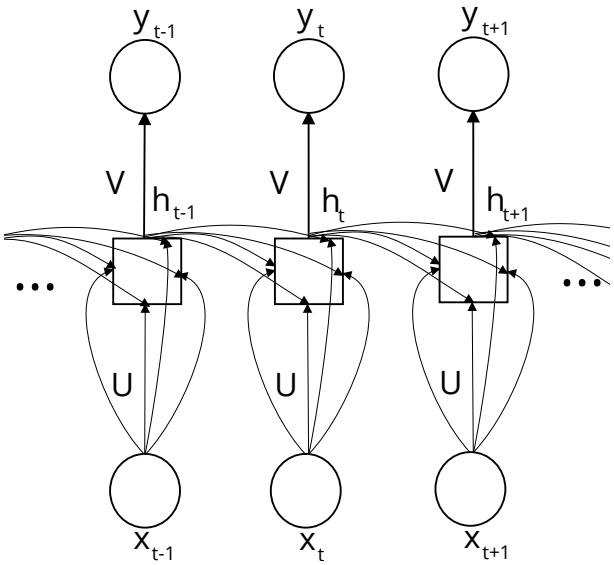

Figura 15 – Rede LSTM

A arquitetura de uma LSTM se assemelha à de uma Vanilla RNN (Figura 13), porém no caso da LSTM, o seu estado interno consiste de um conjunto de sub-redes conectadas recorrentemente chamadas bloco de memória. Esses blocos contém células de memórias com auto conexões capazes de armazenar o estado temporal da rede, além de unidades especiais chamada portas (*gates*) que são responsáveis por controlar o fluxo de informações, como ilustrado

na Figura 16. Cada bloco consiste em uma ou mais células de memória conectada com três portas: esquecimento (*forget*, Eq. 2.1), entrada (*input*, Eq. 2.2) e saída (*output*, Eq. 2.3). Cada porta tem uma função que permite redefinir, escrever e ler as operações dentro do bloco de memória. Cada *gate*, usa uma função logística sigmoide (σ) para achatar os valores desses vetores entre 0 (porta fechada) e 1 (porta aberta). A rede só pode interagir com as células através das portas. Para as equações seguintes, as variáveis utilizadas, i_t, f_t , o_t , c_t, h_t, x_t,y_t, h_t são vetores que representam valores no tempo t. W_* são matrizes de peso conectado a diferentes portas, e b_* são os vetores que correspondem ao *viés* (bias) e $\odot$ representa uma multiplicação elemento a elemento.

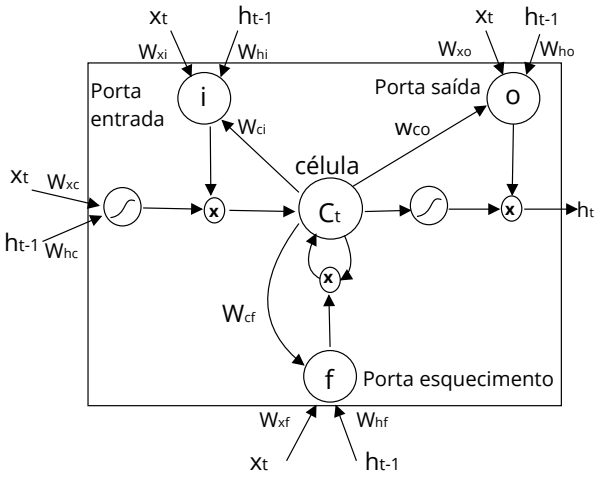

Figura 16 – Bloco de memória da LSTM. Adaptado de Graves (2012a)

$$f_t = \sigma(W_{xf} * x_t + W_{hf} * h_{t-1} + W_{cf} * c_{t-1} + b_f) \tag{2.1}$$

$$i_t = \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + W_{ci} * c_{t-1} + b_i) \tag{2.2}$$

$$o_t = \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + W_{co} * c_t + b_o) \tag{2.3}$$

$$\tilde{C}_t = tanh(W_{xc} * x_t + W_{hc} * h_{t-1} + b_c) \tag{2.4}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{2.5}$$

A porta *forget* (f_t) define se as ativações do estado anterior serão aproveitadas na memória, de maneira que caso as características detectadas sejam importantes a porta será aberta e as informações contidas poderão ser carregados para os próximos intervalos de tempo, caso contrário o conteúdo da célula de memória anterior (C_{t-1}) será redefinido.

A porta de entrada (i_t) define o quanto do novo estado calculado para a entrada corrente será aproveitado, e por fim, a porta de saída (o_t) define se o estado interno será exposto para o resto da rede (rede externa). Em seguida, uma camada tangente hiperbólica cria um vetor de novos valores $\tilde{C}_t$ (Eq. 2.4) dito como candidatos para serem adicionados no novo estado da célula de memória.

A unidade de memória interna C_t é uma combinação da memória anterior C_{t-1} multiplicada pela porta de esquecimento e os valores candidatos $\tilde{C}_t$ multiplicado pela porta de entrada (Eq. 2.5). Intuitivamente, percebe-se que a memória é uma combinação da memória no tempo anterior com a nova no tempo corrente.

Dado a memória C_t, finalmente é possível calcular a saída do estado oculto h_t multiplicando as ativações da memória com a porta de saída (Eq. 2.6).

$$h_t = o_t \odot tanh(C_t) \tag{2.6}$$

A Figura 17 exemplifica como LSTM preserva o gradiente, contornando o problema do desaparecimento do gradiente exibido na Figura 14. As portas de entrada, esquecimento e saída são respectivamente exibidas abaixo, à esquerda e acima do bloco de memória. A célula de memória lembra as primeiras entradas, desde que a porta de esquecimento esteja aberta e a porta de entrada esteja fechada. Quando a porta de esquecimento e de entrada estão abertas há uma mistura de informações de tempos anteriores com o corrente. A porta de saída fornece um controle de fluxo de informação para a camada de saída.

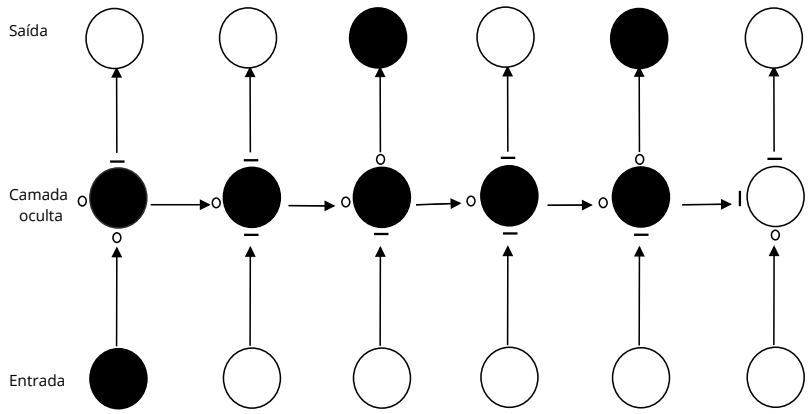

Figura 17 – Resolução do problema do desaparecimento do gradiente utilizando LSTM. Adaptado de Graves (2012a)

No entanto, LSTM só é capaz de fazer uso da informação de contexto anterior. Na tarefa de modelagem sequencial, podemos ter acesso a recursos de entrada do passado e futuro por um determinado tempo, para isso uma extensão da LSTM foi criada por Graves e Schmidhuber (2005) e nomeada Bidirecional LSTM (BLSTM) (Figura 18). Sua arquitetura é formada por

um empilhamento de duas camadas intermediárias separadas, consistindo de uma sequência *forward* e *backward* que são transmitidos para a mesma camada de saída, fazendo então, uso da informação contextual de ambos os lados da sequência.

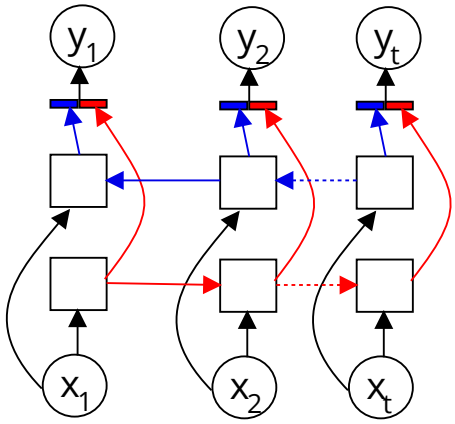

Figura 18 – Arquitetura Bidirecional LSTM (BLSTM). Adaptado de Wang et al. (2015)

2.4 Classificação Temporal Conexionista

Normalmente, RNNs são restritas a problemas onde haja o alinhamento entre a sequência de entrada e saída sejam desconhecidos (GRAVES, 2012b), ou seja, para cada intervalo de tempo na sequência de entrada existe um rótulo correspondente. Para realizar o treinamento em dados não alinhados, utiliza-se a Classificação temporal conexionista, do inglês *Connectionist Temporal Classification* (CTC) (GRAVES et al., 2006) (GRAVES, 2012b), uma função genérica de custo que permite o treinamento de sequências em que o alinhamento entre a entrada e saída sejam desconhecidos (Figura 19), representando assim, uma sincronização temporal da sequência de saída em relação à sequência de entrada. CTC converte uma sequência de rotulagem com informação temporal em uma sequência mais curta de rótulos removendo informações de sincronismo e alinhamento.

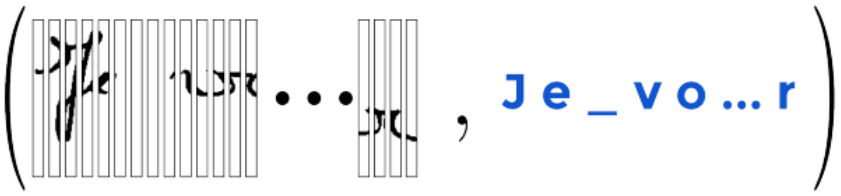

Figura 19 – Exemplo da aplicação do CTC onde não há alinhamento temporal, adaptado de Bluche (2015)

O treinamento consiste de um conjunto de exemplos S, onde cada elemento é um par de sequências $(x;y)$. A sequência $x = (x_1, x_2, ..., x_T)$ é uma sequência de entrada de tamanho T pertencente ao espaço de entrada $\mathscr{X}$, enquanto y = $(y_1, y_2, ..., y_U)$ é a sequência de saída

esperada de tamanho U pertencente ao espaço de saída $\mathscr{Y}$, onde $U \leq T$. Para a utilização do CTC é necessário estender o alfabeto de rótulos $\overline{\mathscr{Y}}$ como $\mathscr{Y} \cup \phi$, onde ϕ denota um rótulo "vazio", ou seja, a sequência $(y_1, \phi, \phi, y_2, \phi, y_3)$ é equivalente a (y_1, y_2, y_3). A rede então gera uma distribuição de probabilidade sobre o espaço de todos os possíveis rótulos pertencentes a $\overline{\mathscr{Y}}$. Juntas, essas probabilidades estimam uma distribuição sobre os elementos $\pi \in \overline{\mathscr{Y}}$, onde π é conhecido como "caminho", uma saída gerada pela rede.

A sequência de saída z é representada por $z = \beta(\pi)$. Onde β mapeia caminhos π para o conjunto $\mathscr{Y}$ de possíveis rótulos. Isso é feito primeiramente removendo rótulos repetidos consecutivos e em seguida, removendo os rótulos "vazios" dos caminhos. Por exemplo, $\beta(\phi \text{aa} \phi \phi \text{abb})$ = $\beta(\text{a} \phi \text{ab} \phi)$ = aab. CTC emprega o algoritmo *forward-backward* similar ao empregado no treinamento do HMM (RABINER, 1989) para calcular o gradiente da função de perda.

Em seguida, é realizado a decodificação (Figura 20), onde é possível rotular uma sequência de dados de entrada x desconhecidos escolhendo a rotulação mais provável em l*:

$$l* = \arg\max_{l} p(l|x) \tag{2.7}$$

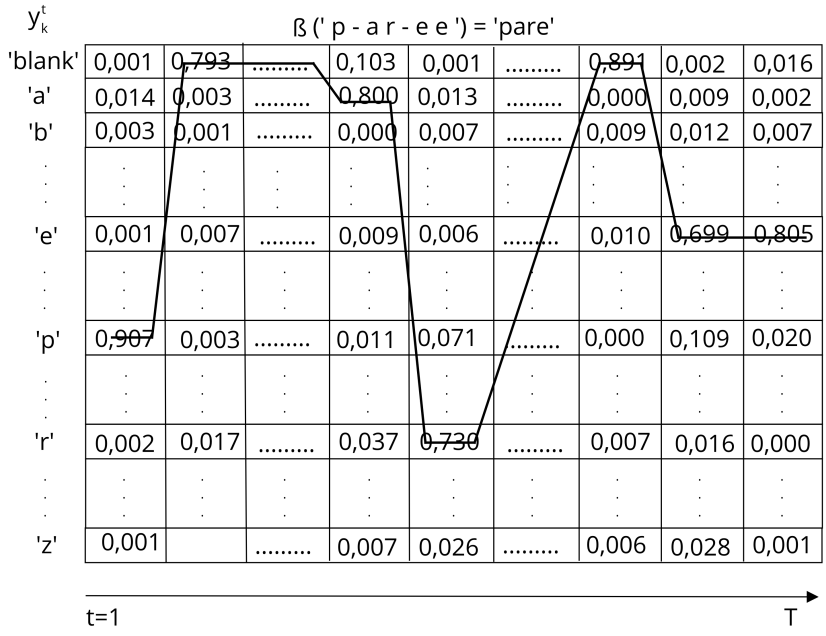

Figura 20 – Exemplo da decodificação, selecionando o rótulo com máxima probabilidade.

A tarefa de classificação acima é tido como um desafio pois um pequeno erro levará a classificação errada da palavra (LIPTON; BERKOWITZ; ELKAN, 2015). Quando o dicionário é conhecido, pode-se restringir a sequência de rotulagem de saída de acordo com um léxico. Para isso, existe uma abordagem baseada na Distância de Levenshtein, do inglês *Levenshtein Distance* (LD) (LEVENSHTEIN, 1966), que compara o rótulo y com as palavras contidas no dicionário através de uma distância de edição, onde é possível transformar uma sequência de caracteres em outra. O funcionamento do algoritmo consiste em determinar o número mínimo

de inserções, substituições e remoções para que uma cadeia de caractere seja igual a outra. A distância de edição pode ser definida da seguinte forma: a distância $d(y,z)$ entre duas strings y e z é o menor custo de uma sequência de operações que transforma y em z. O custo de uma sequência de operações é a soma dos custos de operações individuais.

3

Materiais e Métodos

Neste capítulo, apresenta-se o material utilizado e os procedimentos realizados no desenvolvimento da pesquisa.

3.1 Métodos

Para realização desta pesquisa buscou-se compreender qual a influência de diferentes tipos de ruídos aditivos no desempenho de sistemas automáticos de reconhecimento de fala. Para isso, foram conduzidos experimentos que envolvem o reconhecimento de palavras isoladas com ruídos aditivos. Após a aquisição do áudio, na etapa de extração de características foi gerado um vetor de 600 dimensões. Este vetor corresponde a uma janela de contexto composta por 15 *frames*, em que cada um corresponde a um vetor de 40 dimensões (Figura 21). Cada componente de um *frame* é a energia do sinal acústico de duração de 25 ms em um dos 40 filtros na escala mel. O *frame* central é aquele empregado na classificação. Os sete *frames* adjacentes, à esquerda e à direita, formam o contexto. Nesta janela, os *frames* não cobrem eventos distintos, isto é, existe um entrelaçamento. Eles estão sobrepostos em intervalos de duração de 10ms. A ideia empregada neste processo foi proposta por Abdel-Hamid et al. (2014) e adotado também por Santos et al. (2015).

Em seguida, os modelos preditivos foram treinados com a base limpa, isto é, sem adição de ruídos, e os testes foram realizados com inclusão dos ruídos aditivos provenientes da base *NOISEX-92*[1] (VARGA; STEENEKEN, 1993). Estes ruídos foram: conversa (gravação feita com 100 pessoas falando em uma cantina com raio de 2 metros), volvo (gravação feita dentro do veículo volvo a 120km/h em uma estrada de asfalto em condições de chuva) e fábrica (ruído gravado em uma sala de produção de automóveis). Esses ruídos foram aplicados na base limpa mantendo uma relação sinal-ruído, do inglês *signal-to-noise ratio* (SNR) de 6dB (SANTOS et

[1] <http://spib.linse.ufsc.br/noise.html>

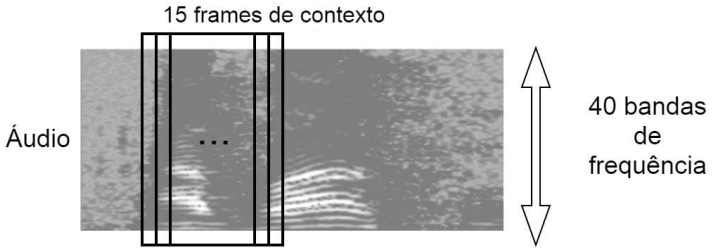

Figura 21 – Extração de coeficientes dimensão 40 por 15 *frames* de contexto. Adaptado de Santos et al. (2015)

al., 2015). Assim como em Raulino, Duarte e Montalvao (2014), as versões ruidosas do sinal foram obtidas a partir de $sr_k(m) = s_k(m) + r(m)$, onde $r(m)$ representa o sinal de ruído e $s_k(m)$ representa o sinal do áudio sem contaminação de ruído. As bases foram divididas em 90% para treinamento e 10% para teste, seguindo os mesmos critérios de experimentações de Santos et al. (2015).

As abordagens foram avaliadas usando uma série de rodadas de treinamento e testes, onde o conjunto de treinamento é dividido em N subconjuntos. Destes, 1 subconjunto é retirado para teste e o restante para treinamento. Todo o processo é então repetido N vezes, de modo que não haja superposição dos conjuntos de teste (Figura 22). O desempenho é estimado calculando o erro médio ou taxa de acerto médio sobre estes N rodadas. Para Kohavi et al. (1995) o objetivo de múltiplas repetições é aumentar a confiabilidade da estimativa da precisão do classificador.

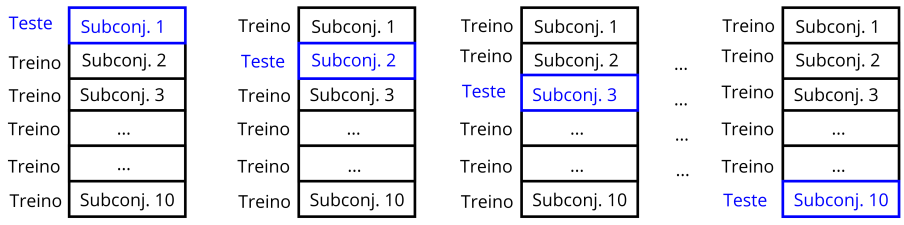

Figura 22 – Funcionamento da bateria de treinamento e testes para 10 subconjuntos

Para verificar se de fato ocorreu uma variação significativa entre o desempenho das máquinas sequenciais, foi empregado um teste estatístico para confrontar a acurácia obtida com HMM e aquelas obtidas com BLSTM, já que a arquitetura bidirecional (BLSTM), e não a mais simples (LSTM), foi a utilizada em todos os experimentos.

A variável aleatória deste teste é a acurácia dos classificadores. As amostras coletadas são pareadas, oriundas do treinamento e teste destes classificadores realizados com os mesmos exemplos em 10 rodadas. Desejamos saber se as médias populacionais são iguais, isto é, se as diferenças nas amostras são meramente fruto do acaso, ou diferentes, caso em que as médias populacionais são distintas. Neste caso como não existe suporte para realização de um teste paramétrico pois, segundo Japkowicz (2011), a robustez do teste t para amostras pareadas requer

amostras com pelo menos 30 realizações, adotamos o teste não paramétrico de Wilcoxon. O nível de significância empregado foi $0,05$ e as hipóteses nula e alternativa são as seguintes:

H_0: A média das populações (acurácia dos classificadores) são iguais.

H_1: A média das populações são diferentes.

Para esse teste, foi utilizado linguagem *R*, que disponibiliza um ambiente de desenvolvimento integrado para cálculos estatísticos e geração de gráficos.

3.1.1 Rotulação automática

A rotulação automática, apresentada através de Algoritmo 1, foi criada com o objetivo de substituir a rotulação fonética utilizada em Santos et al. (2015), realizando o fatiamento do espectrograma em blocos de largura fixa -*n*- em que *n* é um parâmetro fornecido.

A quantidade de blocos é calculada de acordo com o tamanho do espectrograma de cada palavra, isso é feito através de uma divisão entre o tamanho total do espectrograma por *n*, sendo recuperado o valor inteiro dessa divisão. Como exemplo, a Figura 23 possui tamanho total de 450 $frames$, dividindo esse valor por $n = 50$, então serão criados 9 blocos de contexto. O mesmo acontece para espectrogramas de tamanhos menores, como o exemplo da Figura 24, representando a palavra 'recue', neste, são utilizados 5 blocos de contexto de tamanho 50, devido o tamanho total do espectrograma ser de 251.

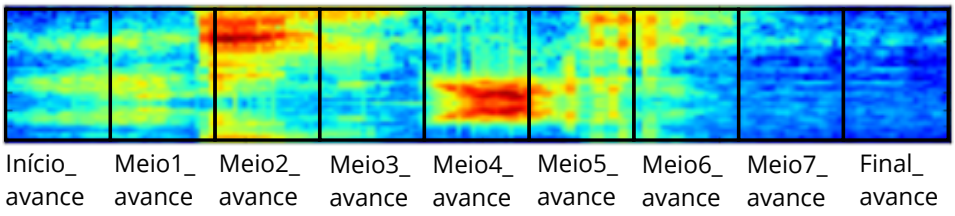

Figura 23 – Exemplo da rotulação automática da palavra avance

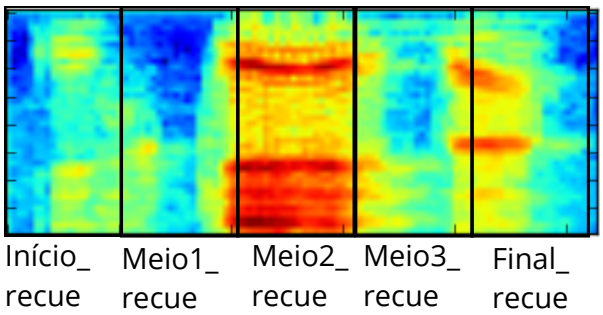

Figura 24 – Exemplo da rotulação automática da palavra recue

Em seguida, são gerados rótulos automaticamente para todos os blocos criados de acordo com cada palavra, ou seja, serão criados rótulos que classificam o início, meio e final. Os rótulos

que definem a sequencia do meio, são classificados de maneira sequencial, de forma que sejam rótulos diferentes. Isto acontece separadamente para todas as palavras presentes na base de dados.

Algoritmo 1: Geração dos blocos de contexto e rotulação automática

```
Entrada: Lista com todas as amostras de uma palavra
Saída: Lista contendo os blocos dos espectrogramas e seus respectivos rótulos
início
    Defina tamanho do bloco;
    repita
        Qtd = Divida o tamanho total do espectrograma da palavra corrente pelo
          tamanho do bloco;
        Inicio = Defina o valor de início do espectrograma (define-se valor 0);
        para i ← 0 até Qtd faça
            Fatie o espectrograma iniciando da posição Inicio até o tamanho do
              bloco mais o valor do Inicio ;
            if i for igual a 0 then
                Este bloco é rotulado como 'inicial';
            else if i for igual a Qtd then
                Este bloco é rotulado como 'final';
            else
                Este bloco é rotulado como 'meio';
            end
        fim
        Inicio recebe o valor de Inicio mais o tamanho do bloco ;
    até a última amostra da lista;
    Defina um rótulo numeral para o bloco que representa o rótulo 'inicial';
    Verifique qual rótulo é referente a posição inicial, meio e final;
    Maior = Percorra a lista e verifique qual amostra tem maior quantidade de
      rótulos 'meio' ;
    repita
        Substitua o rótulo 'inicial' pelo numeral definido ;
        Substitua o rótulo 'final' pelo numeral definido através da variável Maior;
        para i ← 0 até Qtd de blocos por palavra faça
            if rótulo for igual 'meio' then
                Substitua o rótulo 'meio' por um numeral que represente o rótulo
                  'meio' + 'i';
        fim
    até a última amostra da lista;
fim
```

3.2 Materiais

Foram escolhidas duas bases de dados para compor os experimentos. A primeira base utilizada foi a Biochaves [2], criada pelo grupo de pesquisa em biometria do Departamento de

[2] <http://www.biochaves.com/en/download.htm>

Engenharia Elétrica na Universidade Federal de Sergipe (UFS). A base é composta pelas palavras: ‘avance’, ‘direita’, ‘esquerda’, ‘pare’ e ‘recue’ pronunciadas 10 vezes por 8 locutores distintos (6 homens e 2 mulheres), pronunciadas em português brasileiro. As amostras nesta base foram coletadas através de um *smartphone* em ambientes não controlados, como domicílios e salas de aulas a uma taxa de 8000 amostras por segundo com 16-bit de quantização (RAULINO; DUARTE; MONTALVAO, 2014).

A segunda base utilizada é o *corpus* de fala árabe com palavras isoladas que contém 9992 gravações de 20 palavras pronunciadas por 50 locutores homens nativos do idioma árabe. Os arquivos de áudio foram gravados com uma taxa de amostragem de 44.100 Hz e 16 bits de quantização. A base foi construída na Universidade King Faisal, localizado na Arabia Saudita e está disponível de forma gratuita para uso não comercial[3].

Para esse trabalho foram escolhidas 5 palavras para compor os experimentos. A razão é que estas palavras possuem mais de uma sílaba, enquanto que várias outras são monossilábicas. O propósito de utilizar palavras com mais sílabas é para dar um foco maior na máquina sequencial, uma vez que, quanto maior a palavra, maior a estrutura sequencial que a rede irá aprender. As palavras escolhidas foram: ‘um’, ‘dois’, ‘três’, ‘quatro’ e ‘cinco’ em árabe, pronunciadas 10 vezes por 10 locutores. Na Tabela 1 é apresentado os detalhes das palavras escolhidas.

Tabela 1 – Detalhamento das palavras escolhidas, adaptado de Alalshekmubarak e Smith (2014)

Palavra em árabe	**Palavra em Português**	**Pronúncia em inglês**
وَاحد	Um	Wahed
اثنَان	Dois	Ethnan
ثلَاثة	Três	Thlatha
أربعة	Quatro	Arbah
خمسة	Cinco	Khamsahs

A primeira coluna descreve a palavra no idioma árabe, a segunda representa a tradução em português e a terceira apresenta a pronúncia em inglês quando falado a palavra em árabe. Os experimentos foram desenvolvidos utilizando a linguagem *Python* 3, e algumas bibliotecas específicas para o Aprendizado de Máquina, como o *Theano, Keras e Numpy*.

A implementação do HMM foi cedida pelos autores Santos et al. (2015), onde foi utilizado um modelo contínuo aplicado o algoritmo de treinamento Baum-Welch. Neste estudo de caso, para cada palavra a ser reconhecida foi criado um modelo HMM, com 16 estados.

[3] <http://www.cs.stir.ac.uk/~lss/arabic/>

4

Experimentos e Resultados

Nesse capítulo serão apresentados e discutidos os principais resultados obtidos no desenvolvimento deste trabalho. Os experimentos foram realizados com a finalidade de avaliar a robustez da abordagem proposta. Para isso, foram utilizadas duas bases de dados: Biochaves e Corpus de fala árabe.

4.1 Pré-Processamento

Esta etapa é responsável pela extração das características do sinal da voz através dos métodos de bancos de filtro na escala mel. Seu funcionamento se assemelha à maneira como o aparelho auditivo humano processa os sinais. Este método foi aplicado em todos os experimentos presentes neste trabalho. Na Figura 25 é apresentado o diagrama genérico aplicado aos experimentos. Essa funcionalidade assim como outros métodos de extração de características podem ser obtidas através de uma biblioteca em *python* chamada *python_speech_features*[1].

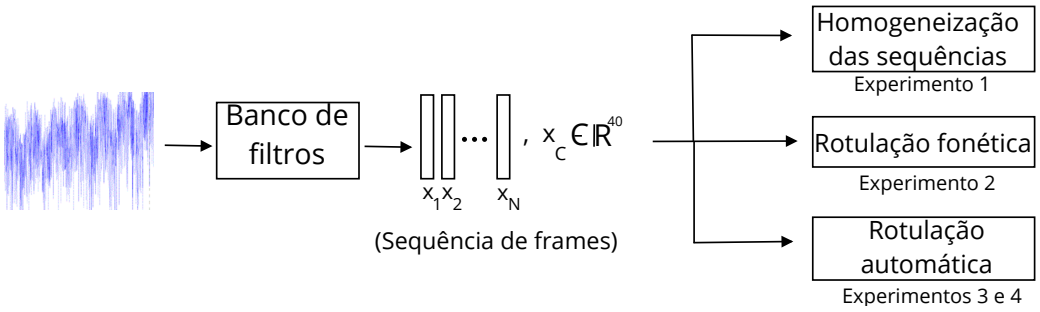

Figura 25 – Pré-processamento dos sinais de áudio

[1] <https://github.com/jameslyons/python_speech_features>

4.2 Métrica

A métrica utilizada para avaliar o desempenho dos experimentos utilizada foi a acurácia. Nestes experimentos, a utilização da acurácia é indicada devido ao fato das classes serem balanceadas, ou seja, existe aproximadamente a mesma quantidade de amostras para todas as classes. Estas taxas foram apresentadas na forma de média $\pm$ desvio padrão, decorrente de 10 rodadas de experimentos. Os valores tabulados nas tabelas de resultados apresentam, portanto, valores médios e dispersão média da acurácia dos diversos classificadores.

A Figura 26 ilustra como são separados as amostras de treinamento e teste. Nesse caso, as amostras de teste não se sobrepõe. São separados 90% para treinamento e 10% para teste em cada rodada, sendo que são 45 amostras para treinamento e 5 para teste para cada locutor. A base Biochaves, é composta por 5 palavras repetidas 10 vezes por 8 locutor, totalizando 360 amostras de treinamento e 40 para teste.

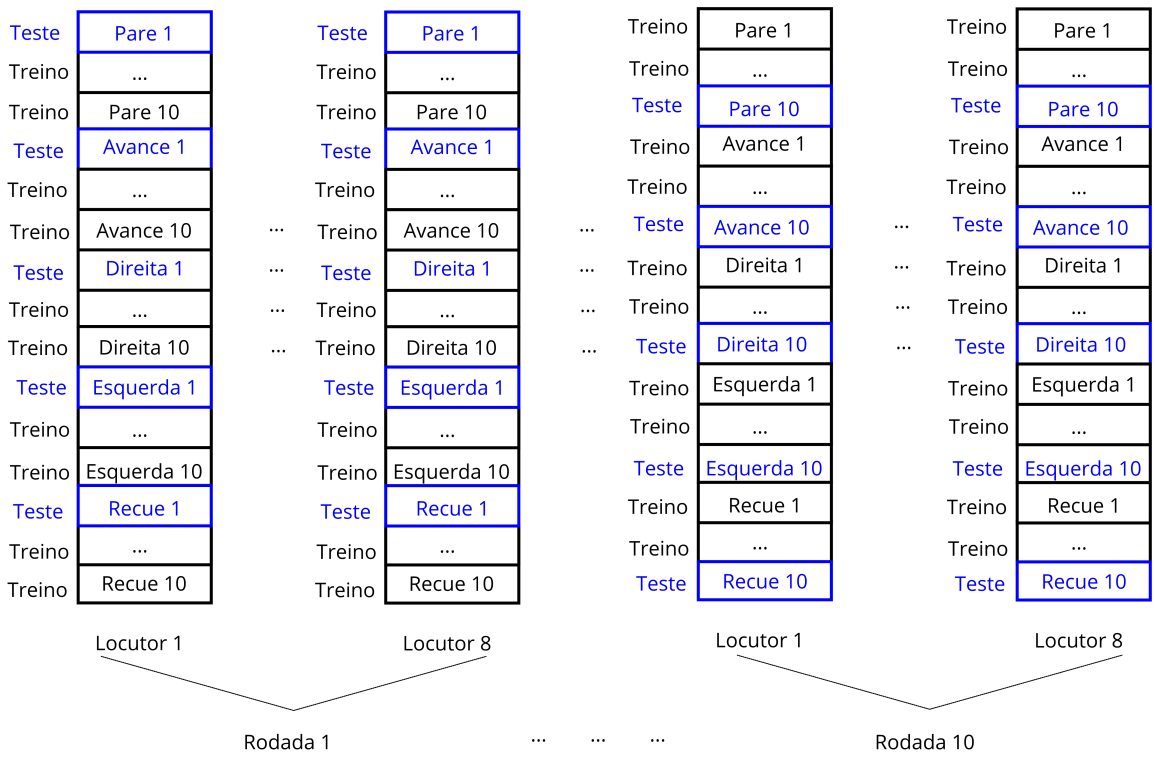

Figura 26 – Detalhamento do processo de treinamento e teste

De forma semelhante, para a base de fala árabe foi utilizado este mesmo processo, porém sua base é composta por 5 palavras repetidas 10 vezes por 10 locutores distintos, totalizando 450 amostras para treinamento e 50 para teste.

4.3 Experimentos com a base BioChaves

Como mencionado na Seção 3.2, a base de dados Biochaves é formada por 5 palavras: ‘avance’, ‘direita’, ‘esquerda’, ‘pare’ e ‘recue’ pronunciadas 10 vezes por 8 locutores distintos em português brasileiro. Para compor o processo de treinamento foram separados 90% da base de dados não contaminada com ruído e os 10% restantes para teste utilizando as bases de dados contaminadas com ruídos aditivos.

4.3.1 Experimento sem rotulação fonética (Experimento 1)

O primeiro experimento (Tabela 2) consistiu no emprego da máquina de aprendizado CNN, BLSTM e HMM de forma isolada. Nesse contexto, a CNN analisa o padrão do espectrograma como uma imagem, enquanto BLSTM e HMM analisam a estrutura sequencial dos *frames* da voz.

Antes de realizar o treinamento dos modelos, foi feito o redimensionamento das matrizes que representam o espectrograma da palavra. Para isso, foram adicionados colunas com valor zero, de modo que todas as imagens de entrada tenham mesma dimensão. Alguns testes foram conduzidos para determinar a melhor configuração para os modelos, e a melhor encontrada da CNN foi composta por 4 camadas convolucionais e subamostragem. Na camada de convolução foi utilizado *kernels* de dimensão 3x3 e na de subamostragem filtro de dimensão 2x2. Foram utilizados mapas de características de tamanhos: 32, 32, 64, 64 respectivamente, em cada camada de convolução. Por fim, a MLP contém 512 neurônios na camada oculta e 5 neurônios na camada de saída, onde cada neurônio representa uma palavra.

Seguindo os mesmos padrões do experimento com CNN, antes de realizar o treinamento da BLSTM, foi realizado o redimensionamento das matrizes para a maior dimensão da matriz representando o espectrograma da maior palavra. Durante a fase de treinamento foi utilizada 512 neurônios na camada oculta, contendo 1 bloco de memória em cada e 5 neurônios na camada de saída, onde cada neurônio representa uma palavra. A taxa de aprendizagem definida foi de 10^{-3}. Estes parâmetros foram definidos baseado em trabalhos como o de Graves, Jaitly e Mohamed (2013).

A Figura 27, ilustra este processo de treinamento, cujo objetivo é verificar o potencial desses modelos isoladamente. Os experimentos foram desenvolvidos por meio do *framework Keras*[2] uma API de desenvolvimento em *python* para redes neurais e a implementação do HMM foi cedida pelos autores Santos et al. (2015)

Os resultados exibidos na Tabela 2, mostra a dificuldade de aprendizado dessas máquinas em funcionamento isolado resultando em taxas de erro de até 57% nas bases de teste.

[2] <https://keras.io/>

Tabela 2 – Experimento 1 - CNN, BLSTM e HMM sem rotulação fonética

Métodos	Acurácia (%)			
	Limpa	Fábrica	Conversa	Volvo
BLSTM	23,4976 % ±5,814	23,1751% ±6,383	23,810 % ±5,678	23,754 % ±5,974
CNN	20,228 % ±5,818	20,228 % ±5,818	20,228 % ±5,818	20,228 % ±5,818
HMM	56,104 % ±10,101	45,865 % ±9,638	41,722 % ±10,806	57,711 % ±7,545

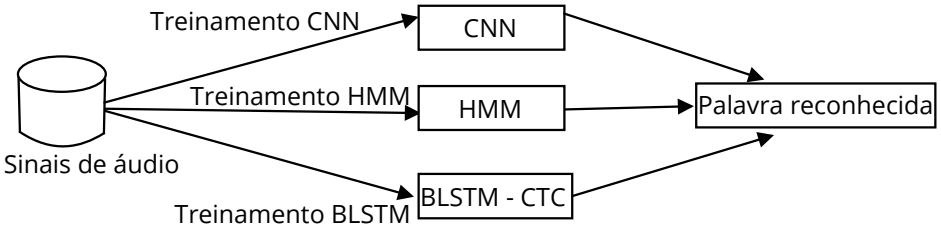

Figura 27 – Processo de treinamento com CNN, BLSTM e HMM

4.3.2 Experimentos com rotulação fonética (Experimento 2)

O próximo experimento baseia-se na hipótese tratada na Seção 1.3, onde sugerimos que a união de duas máquinas de aprendizado possam reduzir as taxas de erro durante a classificação. Nesse contexto, utilizamos a rede CNN para realizar a classificação fonética através de janelas de tempo com largura de 15ms, tempo no qual o sinal da fala é presumivelmente estacionário. As sequências de rótulos emitidos pela máquina CNN são classificados subsequencialmente por uma máquina sequencial.

A rotulação fonética foi definida sob supervisão humana nas bases de áudio, através de um *pluging* chamado *EasyAlign* para o *software* Praat[3]. A rotulação foi feita em todas as amostras da base, totalizando 400 amostras. Para cada áudio, foram mapeados os fonemas e a duração de tempo (em milissegundos) de cada um. Nesse processo foram utilizados 15 classes de fonemas (Tabela 3), e mais uma classe que representa *frames* de silêncio. O detalhamento dos fonemas para cada palavra encontra-se na Tabela 4.

Tabela 3 – Rótulos de fonemas utilizados na transcrição fonética das palavras

Fonemas														
/a/	/v/	/an/	/c/	/i/	/d/	/r/	/ec/	/t/	/s/	/k/	/rr/	/p/	/k/	/u/

A CNN foi composta por duas camadas convolucionais e subamostragem. Na camada de convolução utilizou-se *kernels* de tamanho 3x3 seguido pela camada de subamostragem com filtro de tamanho 2x2. Na primeira camada foram usados 20 mapas de características e na segunda 50. Por fim, a última camada foi formada por uma *MLP* com 250 neurônios de entrada, 200 na camada oculta e 16 neurônios de saída, onde cada neurônio corresponde a um fonema, servindo como entrada para o treinamento do modelo sequencial. Para a abordagem utilizando HMM, para cada palavra foi criado um modelo diferente. Para a abordagem utilizando LSTM e

[3] <http://www.fon.hum.uva.nl/praat/>

Tabela 4 – Detalhamento por palavra das sub-unidades acústicas utilizadas na transcrição fonética

Palavras	**Fonemas**
avance	/a/ /v/ /an/ /c/ /i/
direita	/d/ /i/ /r/ /ec/ /i/ /t/ /a/
esquerda	/i/ /s/ /k/ /ec/ /rr/ /d/ /a/
pare	/p/ /a/ /r/ /i/
recue	/rr/ /e/ /k/ /u/ /i/

BLSTM, o treinamento consistiu em 256 unidades ocultas com um bloco de memória em cada e uma taxa de aprendizado de 10^{-3}, estes parâmetros foram definidos baseado nos trabalhos de Santos et al. (2015) e Graves (2012b). A Figura 28 ilustra o processo de treinamento de forma genérica e a Figura 29 ilustra o processo de teste.

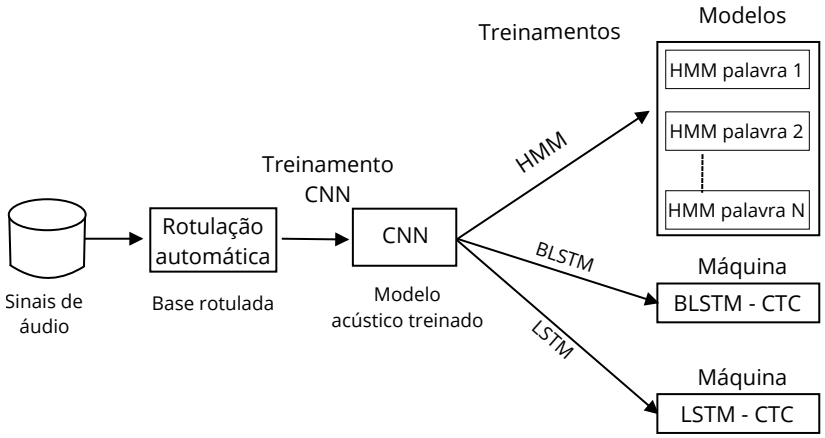

Figura 28 – Processo de treinamento com rotulação fonética utilizando CNN, HMM, LSTM e BLSTM

Neste experimento foram comparadas três máquinas sequenciais: HMM, LSTM e BLSTM. O emprego de LSTM (unidirecional) foi feito com o propósito de verificar o quanto a informação de tempos futuros, utilizada no treinamento e teste de BLSTM, colabora para a melhoria dos resultados. Os modelos HMM foram baseados no trabalho base, elaborado por Santos et al. (2015). Na Tabela 5 é possível verificar que a abordagem utilizando CNN-BLSTM obteve desempenho superior à abordagem CNN-HMM e CNN-LSTM independente do tipo de ruído, além de fornecer resultados mais estáveis, tendo em vista que a variância do erro é menor.

Tabela 5 – Experimentos 2 - Rotulação fonética nas bases de áudio

Métodos	Acurácia (%)			
	Limpa	Fábrica	Conversa	Volvo
CNN-BLSTM	**100**%	**92,688**% $\pm 6,10$	**85,009**% $\pm 4,91$	**100**%
CNN-HMM	91,679% $\pm 25,42$	80,903 % $\pm 23,51$	76,120% $\pm 20,95$	91,679 % $\pm 25,42$
CNN-LSTM	85,652% $\pm 23,903$	68,708 % $\pm 19,997$	62,486% $\pm 16,091$	85,91 % $\pm 24,104$

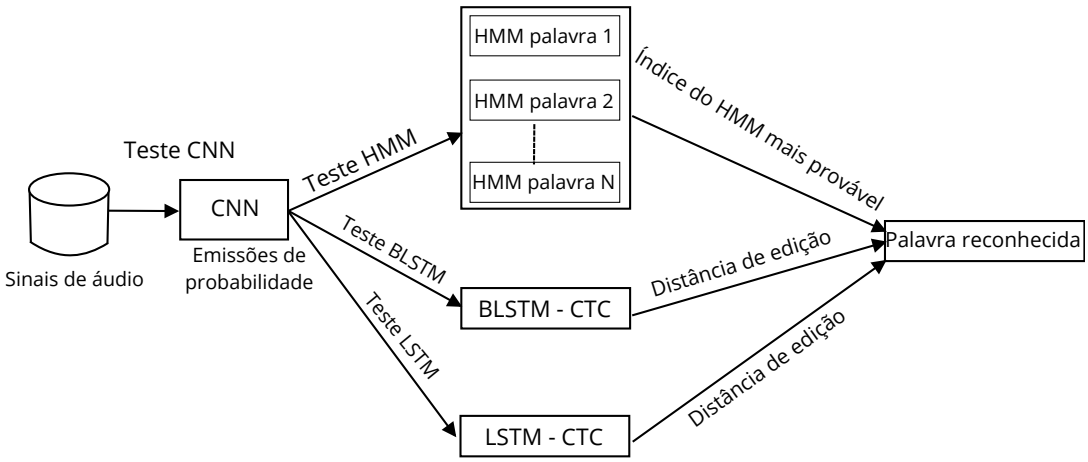

Figura 29 – Processo de teste com rotulação fonética utilizando CNN, HMM, LSTM e BLSTM

Para validar este experimento, foi realizado o teste estatístico não paramétrico de Wilcoxon pareado utilizando os resultados do CNN-BLSTM e CNN-HMM para cada base: Limpa, Conversa, Fábrica e Volvo, o detalhamento dos resultados são exibidos na Tabela 6.

Tabela 6 – Resultados detalhados das 10 rodadas utilizando rotulação fonética.

Base	Métodos	1	2	3	4	5	6	7	8	9	10
Conversa	CNN-BLSTM	92,3	89,743	84,615	82,051	87,179	82,051	84,615	92,308	90,625	77,419
	CNN-HMM	87,179	71,795	71,795	77,419	87,179	84,615	87,179	87,179	87,5	19,355
Fábrica	CNN-BLSTM	97,436	87,18	92,308	79,487	97,436	89,743	97,436	92,308	100	93,548
	CNN-HMM	97,436	74,359	74,359	79,487	92,307	87,179	94,872	94,872	96,875	19,355
Volvo	CNN-BLSTM	100	100	100	100	100	100	100	100	100	100
	CNN-HMM	100	100	100	100	100	97,436	100	100	100	19,355
Limpa	CNN-BLSTM	100	100	100	100	100	100	100	100	100	100
	CNN-HMM	100	100	100	100	100	97,436	100	100	100	19,355

O teste de Wilcoxon, apresentado na Tabela 7, nos permite observar que para as bases Conversa e Fábrica ($p-value < 0,05$) a hipótese nula é rejeitada, nesse caso, podemos dizer que, com um grau de confiança de 95%, as populações são diferentes, ou seja, um método pode ser considerado melhor que o outro. Já para as bases Volvo e Limpa ($p-value > 0,05$), a hipótese nula é aceita indicando não haver diferença significativa entre as abordagens envolvidas.

Tabela 7 – Estatística do Teste Wilcoxon para o experimento com rotulação fonética para a base de dados conversa

Base	***p - value***	**Resultado**
Conversa	0.0122	Rejeição de H_0
Fábrica	0.01244	Rejeição de H_0
Volvo	0.1855	Aceitação de H_0
Limpa	0.1855	Aceitação de H_0

Outros resultados obtidos com a mesma base de dados, pode ser encontrado no trabalho

de Santos et al. (2015), apresentado na Tabela 8, onde foram testados outras abordagens acústicas como o SVM e GMM.

Tabela 8 – Experimentos obtidos em Santos et al. (2015)

Métodos	Acurácia (%)			
	Limpa	Fábrica	Conversa	Volvo
SVM-HMM	100,00%	83,20% ± 3,29	77,40% ± 4,22	100,00%
GMM-HMM (5)	99,80% ± 0,42	82,40% ± 4,29	64,20% ± 13,18	99,80% ± 0,42
GMM-HMM (3)	94,50% ± 1,17	76,80% ± 3,42	63,80% ± 11,33	94,50% ± 1,17
GMM-HMM (1)	79,20% ± 2,14	71,40% ± 3,40	57,80% ± 7,39	78,40% ± 1,95

No entanto, anotações fonéticas nem sempre estão disponíveis em bases de dados e sua criação demanda tempo. Na seção 4.3.3 é apresentado uma solução automatizada que elimina o auxílio humano.

4.3.3 Experimentos com rotulação automática (Experimento 3)

Nesse experimento foi utilizada uma abordagem que dispensa a utilização de anotações fonéticas, facilitando o treinamento em bases com grande quantidade de amostras, automatizando todo este processo. Nessa abordagem, após a extração das características, são demarcados blocos de tamanho pré-definidos, de forma que seja aprendido a estrutura sequencial dos blocos para cada palavra, eliminando a supervisão humana desse processo. Para este experimento, Tabela 9, foram utilizados blocos de contexto de tamanho 50. As configurações da CNN, HMM, LSTM e BLSTM foram as mesmas utilizadas no experimento anterior.

Tabela 9 – Experimentos 3 - Rotulação automática nas bases de áudio

Métodos	Acurácia (%)			
	Limpa	Fábrica	Conversa	Volvo
CNN-BLSTM	**95,531**% ±2,90	**74,327** % ±5,77	**72,992**% ±6,44	**95,531** % ±4,62
CNN-HMM	90,769% ±24,74	72,329% ±19,81	71,509% ±18,75	89,687 % ±24,42
CNN-LSTM	85,652% ±23,903	68,708% ±19,99	62,486% ±16,091	85,91 % ±24,104

Os resultados da Tabela 9 para as máquinas CNN-HMM e CNN-BLSTM foram avaliados sob a luz do teste estatístico de Wilcoxon pareado. A acurácia, medida em cada base, está apresentada na Tabela 10.

As diferenças, como observado na Tabela 11, não são significativas, ainda que para a base Volvo, a acurácia média para CNN-BLSTM e CNN-HMM tenham sido muito distintas. Isto se deve ao resultado de uma rodada (quarta rodada) onde a acurácia extremamente baixa da máquina CNN-HMM, diminuiu a acurácia média para esta máquina apresentada na Tabela 9. Esta diferença, no entanto, é fruto do acaso quando avaliada sob a luz do teste de Wicoxon.

Tabela 10 – Resultados detalhados das 10 rodadas dos experimentos utilizando a rotulação automática para a base biochaves

Base	Métodos	1	2	3	4	5	6	7	8	9	10
Conversa	CNN-BLSTM	76,923	82,051	76,923	74,359	71,795	76,923	82,051	71,795	75	87,096
	CNN-HMM	76,923	82,051	74,359	20,513	74,389	71,795	71,795	71,795	84,375	87,096
Fábrica	CNN-BLSTM	71,795	87,179	74,359	75,359	79,472	76,923	82,051	71,795	68,75	70,968
	CNN-HMM	71,795	76,923	87,615	20,513	79,472	89,743	76,923	61,538	78,125	80,645
Volvo	CNN-BLSTM	97,436	100	97,436	87,179	97,436	100	94,872	100	90,625	90,322
	CNN-HMM	97,436	100	100	20,513	97,436	97,436	92,307	94,872	96,875	100
Limpa	CNN-BLSTM	97,436	97,436	94,872	92,308	97,436	97,436	94,872	100	93,75	90,322
	CNN-HMM	97,436	100	100	20,513	97,436	100	94,872	97,436	100	100

Tabela 11 – Estatística do teste estatístico Wilcoxon para o experimento com rotulação automática para a base biochaves

Base	**p-*value***	**Resultado**
Conversa	0.2008	Aceitação de H_0
Fábrica	0.5279	Aceitação de H_0
Volvo	0.4328	Aceitação de H_0
Limpa	0.8255	Aceitação de H_0

A modelagem acústica obtida pela CNN, neste trabalho, é o principal modelo responsável pela robustez ao ruído, o papel da BLSTM assim como HMM é modelar a organização sequencial dos padrões. Nas amostras com o ruído conversa e fábrica, podemos perceber que há uma redução nas taxas de acurácia se comparado as outras bases. No espectrograma da palavra 'avance' nas Figuras 30 e 31 é possível perceber diferentes frequências produzidas pelos diferentes tipos de ruídos. Através dos experimentos é possível perceber que mesmo a abordagem CNN-BLSTM resultando em taxas de acurácia maior e menor desvio padrão se comparado a CNN-HMM, o teste estatístico não paramétrico de Wilcoxon afirma que não há diferença estatística entre essas abordagens.

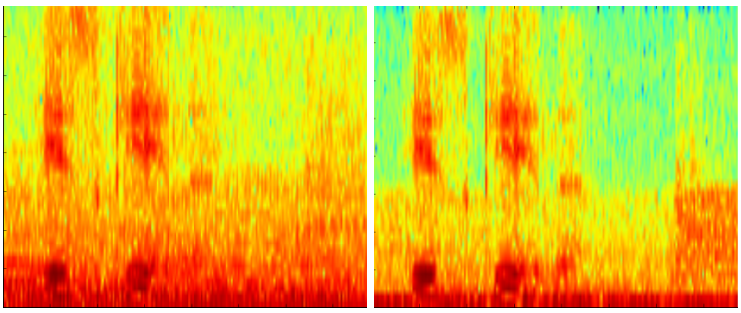

Figura 30 – Espectrogramas representam os ruídos conversa e fábrica respectivamente da palavra avance

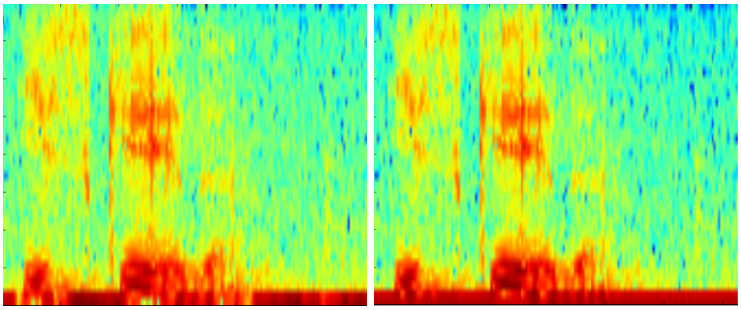

Figura 31 – Espectrogramas representam o ruído volvo e ausência de ruído respectivamente da palavra avance

4.4 Corpus de fala árabe

A base descrita na Seção 3.2 utilizada é o corpus de fala citado em Alalshekmubarak e Smith (2014) com palavras isoladas, onde foram escolhidas 5 palavras em árabe para compor os experimentos, a saber: 'um', 'dois', 'três', 'quatro' e 'cinco', pronunciadas 10 vezes por 10 locutores. Para compor o treinamento foram separados 90% utilizando a base de dados sem ruído aditivo e 10% para teste utilizando bases de dados contaminadas com ruídos aditivos.

4.4.1 Experimentos com rotulação automática (Experimento 4)

Para os experimentos na base de fala árabe como não se dispõe de anotações fonéticas, utilizou-se a abordagem com rotulação automática no nível acústico. Seu funcionamento, assim como também as configurações das máquinas, são as mesmas utilizadas no terceiro experimento da base Biochaves, visto na seção 4.3.3. A Tabela 12 exibe resultados com os áudios a uma taxa de amostragem de 44100Hz (ou 44KHz). No entanto, pode-se perceber o fraco desempenho obtido. Refeito o experimento exibido na Tabela 13 foi reduzido a taxa de amostragem para 8000Hz (ou 8KHz), onde é possível perceber o aumento significativo da acurácia em todas as bases tanto na base limpa quanto com ruídos. Uma possível causa é dada pelo fato de amostras com taxa de 44KHz possuir uma melhor representação do sinal analógico quando convertido para digital, ou seja, há uma maior distribuição de altas frequências espalhadas pelo espectrograma. Quando a taxa é reduzida para 8KHz, há uma redução da qualidade do áudio, porém ainda assim apresenta uma boa representação do sinal, destacando as altas intensidades de frequências como podemos visualizar nas Figuras 32 e 33.

Tabela 12 – Experimentos com rotulação automática base árabe com taxa de amostragem de 44100Hz

Etapa	Acurácia (%)			
	Limpa	Fábrica	Conversa	Volvo
CNN-BLSTM	**27,2** % $\pm 6,31$	**22,4**% $\pm 6,31$	**20**% ± 0	**22,6**% $\pm 6,93$
CNN-HMM	22,6 % $\pm 11,11$	19% $\pm 7,84$	18,2% $\pm 6,42$	19,2% $\pm 7,78$

Tabela 13 – Experimentos com rotulação automática base árabe a uma taxa de amostragem de 8000Hz

Etapa	Acurácia (%)			
	Limpa	Fábrica	Conversa	Volvo
CNN-BLSTM	**95,4** % ±1,89	**89,8** % ±4,36	**80,4** % ±5,56	**91,8** % ±3,70
CNN-HMM	90,8 % ±7,25	86,6 % ±8,00	76,2 % ±7,02	88,6 % ±9,47
CNN-LSTM	53,2 % ±26,703	24,8 % ±4,849	21,8 % ±6,052	24,4 % ±6,09

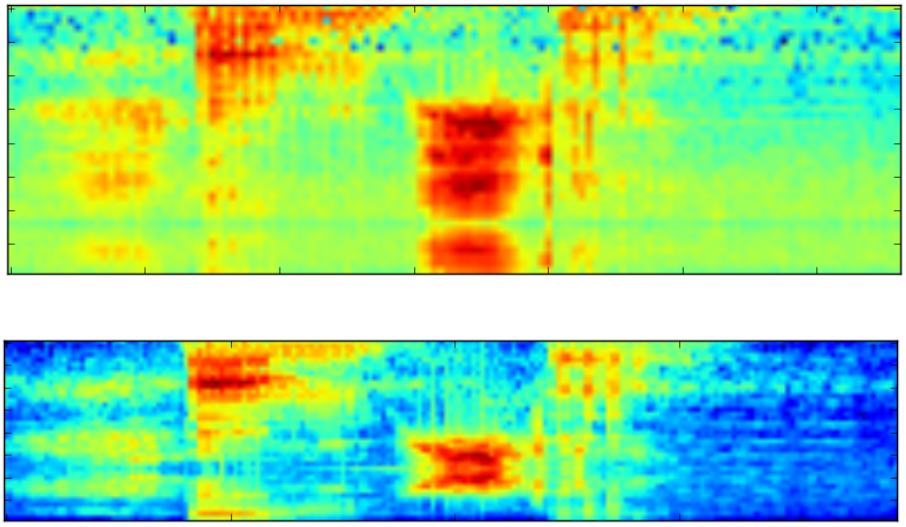

Figura 32 – Espectrograma da palavra **cinco** em árabe com taxa de amostragem de 44100Hz e 8000Hz respectivamente

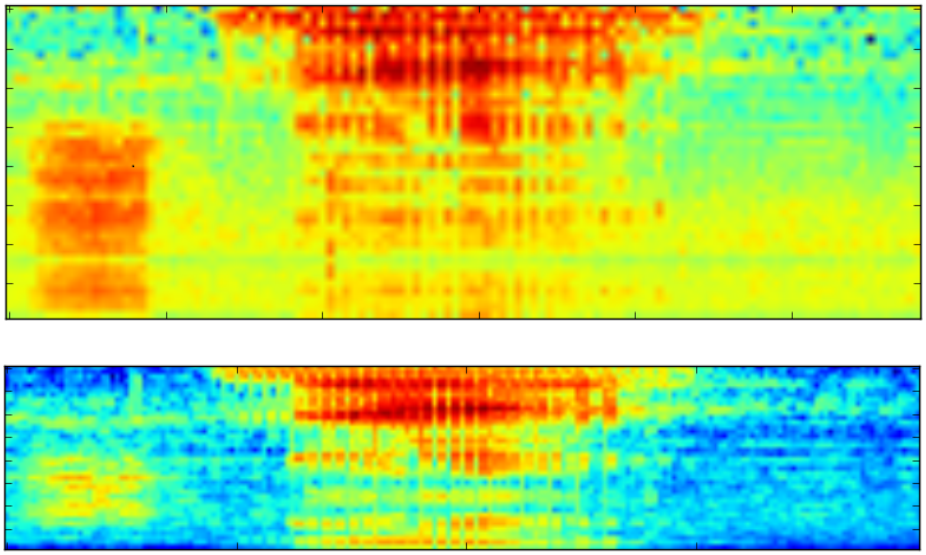

Figura 33 – Espectrograma da palavra **dois** em árabe com taxa de amostragem de 44100Hz e 8000Hz respectivamente

O teste de Wilcoxon utilizou os resultados das 10 rodadas dos experimentos da CNN-

BLSTM e CNN-HMM exibido na Tabela 14. Através do resultado do teste de Wilcoxon, apresentado na Tabela 15, podemos observar que para as bases Limpa, Conversa e Fábrica ($p - value < 0,05$) a hipótese nula é rejeitada, ou seja, um método pode ser considerado melhor que outro. Já para a base volvo ($p - value > 0,05$) indica que não houve diferença significativa entre as abordagens.

Tabela 14 – Resultados obtidos através das 10 rodadas dos experimentos com rotulação automática na base de fala árabe

Base	Métodos	1	2	3	4	5	6	7	8	9	10
Conversa	CNN-BLSTM	82	86	80	84	90	78	78	70	76	80
	CNN-HMM	70	84	82	80	76	80	70	68	68	76
Fábrica	CNN-BLSTM	80	92	90	94	94	90	94	88	90	86
	CNN-HMM	66	92	92	90	86	88	84	90	84	84
Volvo	CNN-BLSTM	86	94	92	94	94	94	96	88	94	86
	CNN-HMM	64	94	94	94	92	96	90	90	82	90
Limpa	CNN-BLSTM	92	94	100	96	96	94	94	96	94	96
	CNN-HMM	74	94	98	90	94	94	86	96	88	94

Tabela 15 – Estatística do teste Wilcoxon para o experimento com rotulação automática na base de fala árabe

Base	**p-value**	**Resultado**
Conversa	0.01191	Rejeição de H_0
Fábrica	0.02108	Rejeição de H_0
Volvo	0.2392	Aceitação de H_0
Limpa	0.01065	Rejeição de H_0

4.5 Discussão Geral

A condição dos experimentos apresentados neste capítulo procurou confirmar a hipótese do trabalho levantada na Seção 1.3. Inicialmente realizamos um experimento cujo objetivo foi verificar o desempenho individual das máquinas CNN, BLSTM e HMM utilizando a base de dados BioChaves. Nesse experimento da Seção 4.3.1 a acurácia auferida foi baixa como o esperado. No entanto, o segundo e terceiro experimento, Seção 4.3.2 e Seção 4.3.3, que empregaram uma composição de máquinas CNN/BLSTM e CNN/HMM, apresentaram resultados com menores taxas de erro.

Ao analisar os resultados obtidos através dos três experimentos, nota-se que a abordagem com rotulação fonética obteve melhores resultados. A rotulação manual, de fato favorece a melhoria dos resultados, pois incorporam o conhecimento do especialista humano. Por outro lado, a abordagem com rotulação automática possui a vantagem de eliminar a supervisão humana o que facilita o treinamento principalmente em base de dados com grande quantidade de amostras para treinamento.

5
Conclusão

Neste trabalho foram avaliadas algumas abordagens referentes ao reconhecimento de fala, com ênfase no problema de reconhecimento de palavras isoladas contaminadas com ruídos aditivos, com característica de ser dependente de locutor em uma base de dados com poucas amostras. Foram utilizadas as máquinas CNN, responsável pela modelagem acústica, HMM para modelagem sequencial, comumente utilizada em sistemas ASR, LSTM e BLSTM, máquinas com arquitetura recorrente capaz de memorizar informações em sequências de longa duração.

Na literatura, existem várias técnicas para a atenuação do ruído, porém, nestes experimentos não foram aplicados nenhum tipo de processamento para diminuição de ruído, ou seja, os dados acústicos serviram diretamente como entrada para os modelos abordados.

Verificamos experimentalmente que os melhores resultados foram obtidos quando utilizado a abordagem com BLSTM. Isto é coerente com o que se discute na literatura, pois, BLSTM é capaz de modelar um contexto arbitrariamente longo através da sua recorrência levando em consideração eventos de tempos passados e futuros, enquanto que HMM modela a estrutura temporal e as dependências apenas entre *frames* adjacentes. É interessante observar que estes resultados foram oriundos de experimentos com bases de dados pequenas, menos de 1000 amostras de treinamento.

Os resultados corroboram a hipótese de pesquisa levantada que afirma que uma composição adequada de máquinas de aprendizado permitiria obter menor erro empírico em função tanto da natureza do padrão de fala, que é localmente estacionário, quanto do balanceamento entre a complexidade do modelo e a disponibilidade de amostras de treinamento, problemática tratada no dilema viés-variância. Ainda nos experimentos também foram comparados diferentes abordagens na etapa da classificação acústica do sinal através de uma abordagem com rotulação fonética e rotulação automática. A melhoria dos resultados do experimento com rotulação fonética sobre a técnica envolvida no experimento com rotulação automática se deve ao auxílio humano uma vez

que o homem é um bom reconhecedor de padrões, em contrapartida exige um maior esforço pois precisa da supervisão humana, ficando inviável sua utilização em bases com grande quantidade de amostras.

5.1 Trabalhos futuros

As principais linhas de investigação que se abriram a partir do ponto em que se encerrou este trabalho são, ao nosso ver, as seguintes:

1. Treinar e testar modelos independente de locutor, sendo utilizados locutores diferentes do treinamento e teste, tornando o reconhecedor capaz de classificar palavras a partir de qualquer locutor.

2. Experimentar outro método de extração de características como Análise Espectral Relativa PLP (RASTA-PLP) apresentado em Alalshekmubarak e Smith (2014).

3. Experimentar outros métodos de modelo sequencial como o *Echo State Networks* (ESN), o que permitiria comparar diretamente resultados obtidos com modelos investigados nesta pesquisa com outros investigados no levantamento bibliográfico.

Referências

ABDEL-HAMID, O. et al. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, IEEE, v. 22, n. 10, p. 1533–1545, 2014. Citado 2 vezes nas páginas 22 e 36.

ALALSHEKMUBARAK, A.; SMITH, L. S. On improving the classification capability of reservoir computing for arabic speech recognition. In: SPRINGER. *International Conference on Artificial Neural Networks*. [S.l.], 2014. p. 225–232. Citado 4 vezes nas páginas 11, 40, 49 e 53.

ANUSUYA, M.; KATTI, S. K. Speech recognition by machine, a review. *arXiv preprint arXiv:1001.2267*, 2010. Citado na página 18.

BALDI, P. et al. Bidirectional dynamics for protein secondary structure prediction. In: *Sequence Learning*. [S.l.]: Springer, 2000. p. 80–104. Citado na página 29.

BALDI, P.; CHAUVIN, Y. Neural networks for fingerprint recognition. *Neural Computation*, MIT Press, v. 5, n. 3, p. 402–418, 1993. Citado na página 19.

BAUM, L. E.; PETRIE, T. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, JSTOR, v. 37, n. 6, p. 1554–1563, 1966. Citado na página 18.

BENGIO, Y. Deep learning of representations: Looking forward. In: SPRINGER. *International Conference on Statistical Language and Speech Processing*. [S.l.], 2013. p. 1–37. Citado na página 19.

BENGIO, Y.; SIMARD, P.; FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, IEEE, v. 5, n. 2, p. 157–166, 1994. Citado na página 28.

BLUCHE, T. *Deep Neural Networks for Large Vocabulary Handwritten Text Recognition*. Tese (Doutorado) — Université Paris Sud-Paris XI, 2015. Citado 2 vezes nas páginas 9 e 33.

CHEN, J.; CHAUDHARI, N. S. Capturing long-term dependencies for protein secondary structure prediction. In: SPRINGER. *International Symposium on Neural Networks*. [S.l.], 2004. p. 494–500. Citado na página 29.

DAVIS, S.; MERMELSTEIN, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, IEEE, v. 28, n. 4, p. 357–366, 1980. Citado na página 25.

DENG, L.; YU, D. et al. Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, Now Publishers, Inc., v. 7, n. 3–4, p. 197–387, 2014. Citado na página 19.

ECK, D.; SCHMIDHUBER, J. A first look at music composition using LSTM recurrent neural networks. *Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale*, v. 103, 2002. Citado na página 29.

ELMAN, J. L. Finding strsdsucture in time. *Cognitive science*, Elsevier, v. 14, n. 2, p. 179–211, 1990. Citado 4 vezes nas páginas 9, 17, 28 e 29.

EYBEN, F. et al. From speech to letters-using a novel neural network architecture for grapheme based asr. In: IEEE. *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*. [S.l.], 2009. p. 376–380. Citado na página 23.

FUKUSHIMA, K. Neocognitron–a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *NHK*, NHK, n. 15, p. p106–115, 1981. Citado na página 19.

FURUI, S. Digital speech processing, synthesis, and recognition (revised and expanded). *Digital Speech Processing, Synthesis, and Recognition (Second Edition, Revised and Expanded)*, Marcel Dekker, Inc., 2000. Citado na página 16.

GEMAN, S.; BIENENSTOCK, E.; DOURSAT, R. Neural networks and the bias/variance dilemma. *Neural computation*, MIT Press, v. 4, n. 1, p. 1–58, 1992. Citado na página 20.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Citado na página 19.

GRAVES, A. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*, 2012. Citado 5 vezes nas páginas 9, 23, 29, 31 e 32.

GRAVES, A. Supervised sequence labelling. In: *Supervised Sequence Labelling with Recurrent Neural Networks*. [S.l.]: Springer, 2012. p. 5–13. Citado 4 vezes nas páginas 9, 30, 33 e 45.

GRAVES, A. et al. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: ACM. *Proceedings of the 23rd international conference on Machine learning*. [S.l.], 2006. p. 369–376. Citado na página 33.

GRAVES, A.; JAITLY, N.; MOHAMED, A.-r. Hybrid speech recognition with deep bidirectional LSTM. In: IEEE. *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. [S.l.], 2013. p. 273–278. Citado na página 43.

GRAVES, A. et al. Unconstrained on-line handwriting recognition with recurrent neural networks. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2008. p. 577–584. Citado na página 29.

GRAVES, A.; MOHAMED, A.-r.; HINTON, G. Speech recognition with deep recurrent neural networks. In: IEEE. *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. [S.l.], 2013. p. 6645–6649. Citado na página 23.

GRAVES, A.; SCHMIDHUBER, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, Elsevier, v. 18, n. 5, p. 602–610, 2005. Citado 2 vezes nas páginas 17 e 32.

HARRIS, M. D. *Introduction to natural language processing*. [S.l.]: Reston Publishing Co., 1985. Citado na página 18.

HINTON, G. et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, IEEE, v. 29, n. 6, p. 82–97, 2012. Citado na página 19.

HINTON, G. E.; OSINDERO, S.; TEH, Y.-W. A fast learning algorithm for deep belief nets. *Neural computation*, MIT Press, v. 18, n. 7, p. 1527–1554, 2006. Citado na página 19.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural computation*, MIT Press, v. 9, n. 8, p. 1735–1780, 1997. Citado 4 vezes nas páginas 17, 28, 29 e 30.

HUBEL, D. H.; WIESEL, T. N. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, Wiley Online Library, v. 195, n. 1, p. 215–243, 1968. Citado na página 26.

ICHIKAWA, A.; NAKANO, Y.; NAKATA, K. Evaluation of various parameter sets in spoken digits recognition. *IEEE Transactions on Audio and Electroacoustics*, IEEE, v. 21, n. 3, p. 202–209, 1973. Citado na página 25.

JAITLY, N. *Exploring Deep Learning Methods for discovering features in speech signals.* Tese (Doutorado) — University of Toronto, 2014. Citado 4 vezes nas páginas 9, 22, 25 e 26.

JAPKOWICZ, M. S. N. *Evaluating Learning Algorithms: A Classification Perspective*. [S.l.]: Cambridge University Press, 2011. ISBN 0521196000,9780521196000. Citado na página 37.

JORDAN, M. I. Attractor dynamics and parallellism in a connectionist sequential machine. Lawrence Erlbaum Associates, 1986. Citado 3 vezes nas páginas 17, 19 e 28.

JORDAN, M. I. Serial order: A parallel distributed processing approach. *Advances in psychology*, Elsevier, v. 121, p. 471–495, 1997. Citado 2 vezes nas páginas 9 e 28.

KOHAVI, R. et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Ijcai*. [S.l.: s.n.], 1995. v. 14, n. 2, p. 1137–1145. Citado na página 37.

KOLÁŘ, M.; HRADIŠ, M.; ZEMČÍK, P. Deep learning on small datasets using online image search. In: ACM. *Proceedings of the 32nd Spring Conference on Computer Graphics*. [S.l.], 2016. p. 87–93. Citado na página 23.

KUBALA, F. et al. Continuous speech recognition results of the byblos system on the darpa 1000-word resource management database. In: IEEE. *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*. [S.l.], 1988. p. 291–294. Citado na página 18.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015. Citado 3 vezes nas páginas 9, 27 e 29.

LECUN, Y. et al. Handwritten digit recognition with a back-propagation network. In: *Advances in neural information processing systems*. [S.l.: s.n.], 1990. p. 396–404. Citado na página 19.

LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, IEEE, v. 86, n. 11, p. 2278–2324, 1998. Citado 3 vezes nas páginas 9, 26 e 27.

LEE, K.-F. *Automatic speech recognition: the development of the SPHINX system*. [S.l.]: Springer Science & Business Media, 1988. v. 62. Citado na página 18.

LEVENSHTEIN, V. I. Binary codes capable of correcting deletions, insertions and reversals. In: *Soviet physics doklady*. [S.l.: s.n.], 1966. v. 10, p. 707. Citado na página 34.

LI, J. et al. An overview of noise-robust automatic speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, IEEE, v. 22, n. 4, p. 745–777, 2014. Citado na página 23.

LIPTON, Z. C.; BERKOWITZ, J.; ELKAN, C. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015. Citado 3 vezes nas páginas 16, 28 e 34.

LIWICKI, M.; BUNKE, H. Handwriting recognition of whiteboard notes. In: *Proc. 12th Conf. of the Int. Graphonomics Society*. [S.l.: s.n.], 2005. p. 118–122. Citado na página 29.

MIKOLOV, T. et al. Empirical evaluation and combination of advanced language modeling techniques. In: *INTERSPEECH*. [S.l.: s.n.], 2011. p. 605–608. Citado na página 26.

MOHAMED, A.-r.; DAHL, G. E.; HINTON, G. Acoustic modeling using deep belief networks. *Audio, Speech, and Language Processing, IEEE Transactions on*, IEEE, v. 20, n. 1, p. 14–22, 2012. Citado na página 17.

MOHRI, M.; PEREIRA, F.; RILEY, M. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, Elsevier, v. 16, n. 1, p. 69–88, 2002. Citado na página 26.

MOREIRA, L. F. A. d. C. et al. Desenvolvimento de um sistema de reconhecimento de fala. 2012. Citado na página 18.

NARAYANAN, A.; WANG, D. Joint noise adaptive training for robust automatic speech recognition. In: IEEE. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2014. p. 2504–2508. Citado na página 23.

RABINER, L. R. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, IEEE, v. 77, n. 2, p. 257–286, 1989. Citado na página 34.

RAULINO, C.; DUARTE, D.; MONTALVAO, J. Analise de espectro através da deteccao de eventos acusticos elementares no plano tempo-frequência. *Simpósio Brasileiro de Automação Inteligente (SBAI)*, p. 1–16, 2014. Citado 3 vezes nas páginas 18, 37 e 40.

SAK, H.; SENIOR, A.; BEAUFAYS, F. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv:1402.1128*, 2014. Citado na página 17.

SANTOS, R. M. et al. Speech recognition in noisy environments with convolutional neural networks. In: IEEE. *Intelligent Systems (BRACIS), 2015 Brazilian Conference on*. [S.l.], 2015. Citado 13 vezes nas páginas 9, 11, 18, 23, 24, 25, 36, 37, 38, 40, 43, 45 e 47.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. *Neural Networks*, Elsevier, v. 61, p. 85–117, 2015. Citado 2 vezes nas páginas 17 e 19.

SELTZER, M. L.; YU, D.; WANG, Y. An investigation of deep neural networks for noise robust speech recognition. In: IEEE. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. [S.l.], 2013. p. 7398–7402. Citado 2 vezes nas páginas 17 e 23.

SONG, W.; CAI, J. End-to-end deep neural network for automatic speech recognition. In: . [S.l.: s.n.], 2015. Citado 2 vezes nas páginas 22 e 29.

STEVENS, S. S.; VOLKMANN, J.; NEWMAN, E. B. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, ASA, v. 8, n. 3, p. 185–190, 1937. Citado na página 25.

SUTSKEVER, I.; MARTENS, J.; HINTON, G. E. Generating text with recurrent neural networks. In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. [S.l.: s.n.], 2011. p. 1017–1024. Citado na página 30.

VARGA, A.; STEENEKEN, H. J. Assessment for automatic speech recognition: Ii. noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems. *Speech communication*, Elsevier, v. 12, n. 3, p. 247–251, 1993. Citado na página 36.

VINTSYUK, T. K. Speech discrimination by dynamic programming. *Cybernetics and Systems Analysis*, Springer, v. 4, n. 1, p. 52–57, 1968. Citado na página 18.

WANG, P. et al. A unified tagging solution: Bidirectional LSTM recurrent neural network with word embedding. *CoRR*, abs/1511.00215, 2015. Citado 2 vezes nas páginas 9 e 33.

WILLIAMS, W. et al. Scaling recurrent neural network language models. *arXiv preprint arXiv:1502.00512*, 2015. Citado na página 26.

WÖLLMER, M. et al. Feature enhancement by bidirectional LSTM networks for conversational speech recognition in highly non-stationary noise. In: IEEE. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. [S.l.], 2013. p. 6822–6826. Citado 2 vezes nas páginas 17 e 29.

Apêndices

APÊNDICE A – Artigo aprovado para publicação na revista IEEE América Latina

Deep Neural Networks for Acoustic Modeling in the Presence of Noise

L. M. Q. D. Santana, R. M. Santos, L. N. Matos and H. T. Macedo

***Abstract*— Systems using deep neural network (DNN) have shown promising results in automatic speech recognition (ASR), where one of the biggest challenges is the recognition in noisy speech signals. We have combined two famous architectures of deep learning, the convolutional neural networks (CNN) for acoustic approach and a recurrent architecture with connectionist temporal classification (CTC) for sequential modeling, in order to decode the frames in a sequence forming a word. Experimental results show that the proposed architecture achieves improved performance over classical models, such as hidden model Markov (HMM) for labeling in variable time sequences in BioChaves database.**

***Keywords*— Speech recognition, Noisy environment, Deep Learning, Convolutional Neural Network, Recurrent Neural Network.**

I. INTRODUÇÃO

A ABORDAGEM dominante para construção de sistemas ASR sempre foi baseada no uso de HMM, para a modelagem da estrutura sequencial da fala, e na Mistura de Gaussianas (GMM) [1] para modelar uma representação espectral da onda sonora. Recentemente, estudos com Redes Neurais Profundas, do inglês *Deep Neural Networks* (DNN), tem apresentado desempenho superior em tarefas de reconhecimento de dados sequenciais [2], [3], [4]. Isto tem motivado sua investigação em tarefas nas quais modelos descritivos, como HMM, eram tipicamente a maneira mais usual de representar uma solução.

DNN é uma técnica de aprendizado de máquina que emprega muitas camadas de processamento de informação e, em alguns modelos, etapas de treinamento supervisionado e não-supervisionado, onde é possível extrair características, analisar e classificar padrões [5].

Um tipo de arquitetura profunda é a Rede Neural Convolucional, do inglês *Convolutional Neural network* (CNN), que é bastante eficaz em tarefas de visão computacional e reconhecimento de imagem. Outro exemplo, é a rede neural recorrente, do inglês *Recurrent Neural Network* (RNN), usada em modelagem de dados sequenciais como voz ou texto. Sua arquitetura profunda é dada através de camadas ao longo do tempo fazendo uso da informação de contexto através de uma conexão recorrente. Entre alguns modelos de RNN, existem a Rede de Jordan [6], Rede Elman [7], e a Memória de Longo Curto Termo, do inglês *Long Short-Term Memory* (LSTM) [8]. Em particular, LSTM possui uma arquitetura capaz de armazenar informações em células de memórias por um período mais longo de tempo e com capacidade de aprender grande quantidade de informações relevantes para a tarefa de regressão ou classificação [9].

Para a tarefa de classificação, normalmente são utilizados algoritmos de aprendizagem de máquina supervisionado, cujo objetivo é ser capaz de produzir uma boa generalização a partir dos dados empregados no treinamento. Isto é, procura-se obter máquinas que produzam uma baixa taxa de erro durante a classificação, quando lhe são apresentados elementos não vistos. Este erro, segundo uma análise realizada em [27], pode ser decomposto através da soma de duas partes conflitantes: a variância e o quadrado do viés. Os autores afirmam que, à medida que se aumenta a complexidade do modelo, isto é, à medida em que a máquina se torna mais complexa, o quadrado do viés tem seu valor diminuído, enquanto que a variância tem seu valor aumentado, sendo o ideal, segundo os autores, encontrar um compromisso entre um modelo complexo (alta variância e baixo viés) e generalizável (baixa variância e alto viés). No problema abordado neste trabalho, admitimos por hipótese que os padrões da fala em curtos segmentos de tempo são estacionários, ou seja, suas propriedades estatísticas são invariantes ainda que este seja um padrão aleatório. Na literatura, alguns autores como [21], [26] e [28] utilizaram janelas de tempo com largura de 15 a 30 ms, pois admitiam existir preservação da estacionariedade do sinal nestes segmentos. No nosso trabalho propomos utilizar uma máquina CNN para classificar elocuções fonéticas, padrões que supostamente são estacionários, e uma máquina sequencial para modelar a estrutura de encadeamento desses padrões. É desta forma que buscamos encontrar o balanço entre viés e variância, tendo em vista que o sistema como um todo é resultado da composição de duas máquinas mais simples. Em particular, exploramos o uso das máquinas de reconhecimento de dados sequenciais HMM e BLSTM com o objetivo de analisar a robustez do sistema quando o sinal de áudio é contaminado com ruído aditivo. Verificamos experimentalmente que os melhores resultados foram obtidos quando utilizado a abordagem com BLSTM devido à sua capacidade de modelar um contexto arbitrariamente longo através da sua recorrência levando em consideração eventos de tempos passados e futuros, enquanto que HMM modela a estrutura temporal e as dependências apenas entre frames adjacentes. Embora este não seja o primeiro trabalho a empregar estas máquinas de aprendizado profundo no

L. M. Q. D. Santana, Universidade Federal de Sergipe (UFS), São Cristóvão, Sergipe, Brasil, lucianamqs@dcomp.ufs.br
R. M. Santos, Universidade Federal de Sergipe (UFS), São Cristóvão, Sergipe, Brasil, rafaelsantos@ufs.br
L. N. Matos, Universidade Federal de Sergipe (UFS), São Cristóvão, Sergipe, Brasil, leonardo@dcomp.ufs.br
H. T. Macedo, Universidade Federal de Sergipe (UFS), São Cristóvão, Sergipe, Brasil, hendrik@dcomp.ufs.br

contexto de reconhecimento da fala, mostramos que a abordagem proposta mostrou-se adequada no reconhecimento de palavras isoladas na presença de ruído, tendo apresentado desempenho superior a outros trabalhos que exploraram a mesma base de dados [14] e [19].

As próximas seções deste artigo estão organizadas da seguinte maneira. A seção II traz uma breve revisão da literatura e trabalhos relacionados. Na seção III, apresentamos a abordagem e base de dados utilizada. Na seção IV, resultados dos experimentos são discutidos. Na seção V, a conclusão do trabalho é apresentada.

II. TRABALHOS RELACIONADOS

O emprego de DNN em tarefas de reconhecimento de fala na presença de ruído foi explorado em diversos trabalhos, a exemplo de [10], [11], [12] e [13]. Em Santos et al. (2015) [14] foi proposto um modelo híbrido composto por CNN-HMM (Fig. 1), onde os autores sugerem que os *frames* do sinal da fala no domínio da frequência possam ser usados como entrada para uma CNN, que por sua vez gera uma sequência de probabilidades usadas pelo HMM, usando o algoritmo de *Baum-Welch*.

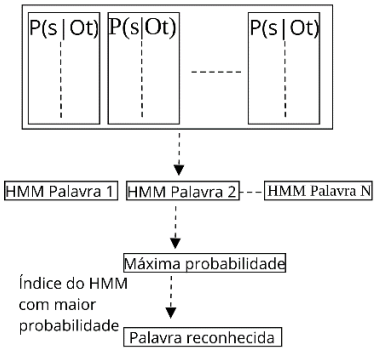

Figura 1. Emissões de probabilidades apresentadas como entrada para HMM.

Em alguns experimentos realizado pelos autores é comparado o uso da GMM, SVM e CNN para a modelagem acústica, onde percebe-se que a CNN foi a menos influenciada por ruídos. Acredita-se que a sua robustez é influenciada pelo compartilhamento de pesos e conectividade local que introduz um certo nível de invariância às distorções presentes em ambientes não controlados. Baseado nisso, foi escolhido a CNN para realizar a modelagem acústica neste trabalho.

Recentemente, a empresa Google anunciou melhorias para transcrição do Google *Voice* usando LSTM. Anteriormente, utilizavam o modelo GMM-HMM, o estado da arte no reconhecimento de fala há mais de 30 anos. Abordagens bem-sucedidas de LSTM podem ser vistas também em [15] e [16], que propõem uma utilização de uma arquitetura recorrente, que combina com Classificação temporal conexionista (CTC, *Connectionist Temporal Classification*) com uma rede recorrente, que prevê cada fonema de acordo com os dados anteriores, obtendo assim um modelo acústico e linguístico de forma conjunta [17]. O autor também demonstra que uma rede profunda LSTM treinada com a regularização adequada consegue alcançar um erro de apenas 17.7% na base TIMIT, (disponível em http://catalog.ldc.upenn.edu/LDC93S1), um *benchmark* de reconhecimento de fala contínua. Esta parece ser a melhor pontuação registrada.

III. MATERIAIS E MÉTODOS

O funcionamento de sistemas ASR é dado primeiramente pela conversão de um sinal acústico produzido pelo homem em um sinal digital de áudio. Podem ser aplicadas algumas técnicas para eliminar os ruídos, períodos de silêncio, etc. Em seguida, é realizada a extração de características, conhecida também como *front-end*, onde o sinal acústico é representado em um espaço de dimensão menor, preservando as informações importantes, ou seja, evidenciando características do sinal que contribuem para a identificação. As características neste trabalho foram extraídas pelo método banco de filtros na escala mel, sendo considerado eficiente para representar as características do sinal da fala que são importantes para a informação do trato vocal. Seus filtros triangulares têm por função discretizar o espectro de frequências de forma similar à maneira como o aparelho auditivo humano os processa [18]. Foram mapeadas para cada elocução, vetores de 40 coeficientes *cepstrais*. A extração foi realizada em frames de 25 milissegundos com 10 milissegundos de entrelaçamento entre eles, sendo este a entrada para a CNN, onde é possível calcular a probabilidade da geração de observações acústicas. Na etapa seguinte, a classificação da sequência temporal de emissões geradas pelas redes convolucionais é feita pela rede BLSTM.

A. *Base de dados*

Os experimentos, assim como em [14] e [19], foram conduzidos envolvendo o reconhecimento de palavras isoladas, pronunciadas em português brasileiro, utilizando a base de dados BioChaves (disponível em http://www.biochaves.com/en/download.htm). A base é composta pelas palavras: ‘avance’, ‘direita’, ‘esquerda’, ‘pare’ e ‘recue’, pronunciadas 10 vezes por 8 locutores distintos (6 homens e 2 mulheres). As amostras foram coletadas através de um *smartphone* em ambientes não controlados, como domicílios e salas de aulas a uma taxa de 8000 amostras por segundo e 16-bit de quantização [19].

Os modelos preditivos foram treinados com a base limpa, isto é, sem ruídos, e os testes foram realizados com inclusão dos ruídos aditivos provenientes da base NOISEX-92 [20], (disponível em http://spib.linse.ufsc.br/noise.html). Estes ruídos foram: conversa (gravação feita com 100 pessoas falando em uma cantina com raio de 2 metros), volvo (gravação feita dentro do veículo volvo a 120km/h em uma estrada de asfalto em condições de chuva) e fábrica (ruído gravado em uma sala de produção de automóveis). Esses ruídos foram aplicados na base limpa mantendo uma relação sinal-ruído (SNR) de 6dB [14].

B. *Arquiteturas utilizadas*

Para o modelo acústico, utilizou-se CNN, cuja arquitetura é composta por sucessivas camadas de convolução e subamostragem realizando um pré-processamento dos dados de entrada e uma MLP (*Multi-Layer Perceptron*) responsável pela saída [14]. Para Song e Cai (2015, p. 2) [21], as CNNs são boas para capturar alto nível de característica no domínio espacial e são invariantes contra as variações nas frequências, no qual é comum observar locutores com tons de voz diferentes. A arquitetura da rede é apresentada através da Fig. 2, onde é possível observar que a rede pode ser formada por uma ou mais camadas de convolução e subamostragem.

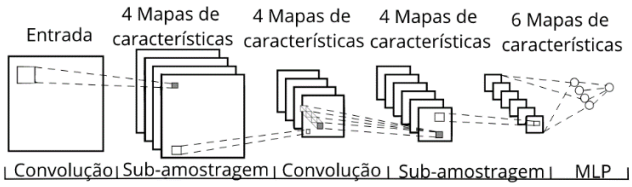

Figura 2. Arquitetura CNN.

A camada convolucional é composta por um conjunto de filtros também conhecidos como *kernel* de convolução. Cada filtro é aplicado a toda imagem de entrada resultando em um mapa de característica (*feature map*), para o caso da voz, o sinal de áudio, que é unidimensional, é transformado para uma representação 2D – o espectrograma – e analisado como uma imagem. No espectrograma, o eixo horizontal é o tempo, tal como no sinal original, enquanto que o eixo vertical é o eixo das frequências, assim pode-se verificar em que bandas a energia se distribui ao longo do tempo em que a fala é pronunciada. Uma propriedade importante nessa etapa é o compartilhamento de pesos entre os *kernels* do mesmo mapa de característica, como o exemplo da Fig. 3. Através disso, é possível detectar padrões independentes de sua localização na imagem de entrada.

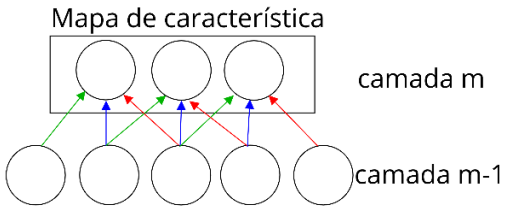

Figura 3. Compartilhamento dos parâmetros para criação de um mapa de características.

Após a convolução, as ativações são passadas para uma segunda camada, a de subamostragem (*subsampling*), onde é possível calcular o valor máximo e médio de uma área de tamanho pré-definido, gerando uma representação de entrada em resolução reduzida. O resultado é um outro mapa de características com resolução menor que proporciona invariação a pequenas translações [22]. Ao término dessas etapas uma MLP calcula a saída final da CNN.

A saída é utilizada como entrada para uma modelo sequencial capaz de modelar uma sequência de fonemas formando uma palavra, através do modelo BLSTM com a função de custo CTC.

LSTM foi inicialmente proposto por Schmidhuber (1997, p. 6) [23], para resolver o tratamento de sequências longas por uma rede recorrente. Na abordagem original, a atualização dos pesos, que é feito por uma variação do algoritmo *Backpropagation*, denominado *Backpropagation Throught Time*, é afetada quando o nível de recorrência é alto, o que impossibilita a rede memorizar sequências de longa duração. Sua arquitetura consiste em um conjunto de sub-redes conectadas recorrentemente chamadas bloco de memória situada na camada oculta. Estes blocos contém células de memórias com auto conexões capazes de armazenar o estado temporal da rede, além de unidades especiais chamado portas (*gates*) que são responsáveis para controlar o fluxo de informações, como ilustrado na Fig. 4. Cada bloco consiste em uma ou mais células de memória conectada com três portas: esquecimento (*forget*, Eq. 1), entrada (*input*, Eq. 2) e saída (*output*, Eq. 3). Cada porta tem uma função que permite redefinir, escrever e ler as operações dentro do bloco de memória. Além de usar uma função logística sigmoide (σ) para achatar os valores desses vetores entre 0 (porta fechada) e 1 (porta aberta).

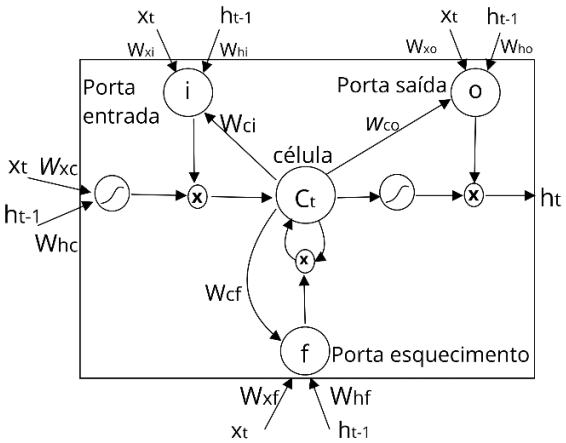

Figura 4. Bloco de memória LSTM.

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (1)$$

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (2)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (3)$$

$$\widetilde{C_t} = tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (4)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (5)$$

A porta *forget* define se as ativações do estado anterior serão aproveitadas na memória. A porta de entrada define o quanto do novo estado calculado para a entrada corrente será aproveitado, e por fim, a porta de saída define se o estado interno será exposto para o resto da rede (rede externa). Em seguida, uma camada tangente hiperbólica cria um vetor de novos valores candidatos $\tilde{C}_t$ (Eq. 4) que poderão ser adicionados na célula. A unidade de memória interna C_t é uma combinação da memória anterior C_{t-1} multiplicada pela porta de esquecimento e os valores candidatos $\tilde{C}_t$ multiplicado pela

porta de entrada (Eq. 5). Intuitivamente, percebe-se que a memória é uma combinação da memória no tempo anterior com a nova no tempo corrente.

Dado essa memória C_t, finalmente é possível calcular a saída do estado oculto h_t multiplicando as ativações da memória com a porta de saída (Eq. 6).

$$h_t = o_t \odot tanh(C_t) \quad (6)$$

As variáveis, i_t, f_t, o_t , c_t, h_t são vetores que representam valores no tempo t. W_* são matrizes de peso conectado a diferentes portas, b_* são os vetores que correspondem ao *viés* (bias) e $\odot$ representa uma multiplicação elemento a elemento.

No entanto, a LSTM convencional só é capaz de fazer uso da informação de contexto anterior, porém, a modelagem da voz é significativamente relacionada com ambos os contextos do passado e futuro. A arquitetura BLSTM (Fig. 5), poder ser criada por um empilhamento de duas camadas intermediárias separadas, consistindo de uma sequência *forward* ($\vec{h}^n$, Eq. 7) e *backward* ($\overleftarrow{h}^n$, Eq. 8) que são transmitidos para a mesma camada de saída fazendo, então, uso da informação contextual de ambos os lados da sequência.

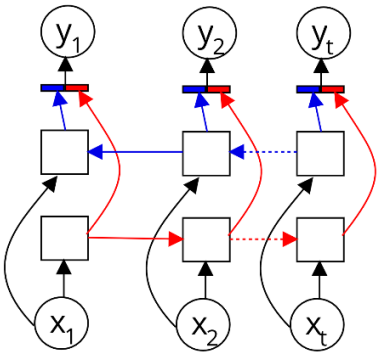

Figura 5. Arquitetura BLSTM.

$$\vec{h}_t^n = \mathcal{H}\left(W_{\vec{h}^{n-1}\vec{h}^n}\vec{h}_t^{n-1} + W_{\vec{h}^n\vec{h}^n}\vec{h}_{t-1}^n + b_{\vec{h}}^n\right) \quad (7)$$

$$\overleftarrow{h}_t^n = \mathcal{H}\left(W_{\overleftarrow{h}^{n-1}\overleftarrow{h}^n}\overleftarrow{h}_t^{n-1} + W_{\overleftarrow{h}^n\overleftarrow{h}^n}\overleftarrow{h}_{t-1}^n + b_{\overleftarrow{h}}^n\right) \quad (8)$$

$$y_t = \left(W_{\overleftarrow{h}^N y}\overleftarrow{h}_t^N + W_{\overleftarrow{h}^n y}\overleftarrow{h}_t^N + b_y\right) \quad (9)$$

Onde, $\mathcal{H}$ é uma função de ativação, geralmente é utilizada a tangente hiperbólica ou sigmoide, e $y = (y_1, y_2, \dots, y_T)$ uma sequência que representa a saída.

Normalmente, RNNs são restritas a problemas onde o alinhamento entre a sequência de entrada e saída sejam conhecidas [24], ou seja, para cada intervalo de tempo na sequência de entrada exista um rótulo correspondente. Para treinar a rede diretamente em dados não alinhados, utiliza-se CTC, uma função genérica de custo que permite o treinamento de sequências em que o alinhamento entre a entrada e saída sejam desconhecidos, representando assim, uma sincronização temporal da sequência de saída em relação à sequência de entrada.

Seja $x = (x_1, x_2, \dots, x_T)$ uma sequência de entrada de tamanho T pertencente ao espaço de entrada X. Dado $y = (y_1, y_2, \dots, Y_u)$, uma sequência de saída de tamanho U pertencente ao espaço de saída Y, para a tarefa de rotulação de sequência x_t representa uma sequência de fonemas e y_t uma sequência que representa uma palavra. Para a utilização do CTC é necessário estender o alfabeto de rótulos $\bar{Y}$ como $Y \cup \phi$, onde ϕ denota um rótulo 'vazio', ou seja, a sequência $(y_1, \phi, \phi, y_2, \phi, y_3)$ é equivalente a (y_1, y_2, y_3). A rede então gera uma distribuição de probabilidade sobre o espaço de todos os possíveis rótulos pertencentes a $\bar{Y}$. Mais formalmente, a ativação y_k^t da unidade de saída k no tempo t é interpretada como a probabilidade da observação do rótulo k no tempo t dado o tamanho T da sequência de entrada x. Juntas, essas probabilidades estimam uma distribuição sobre os elementos $\pi \in \bar{Y}$, onde π é conhecido como 'caminho', uma saída gerada pela rede. A probabilidade de um caminho (dado a sua entrada x) é calculada pelo produto de todos os seus elementos:

$$P(\pi|x) = \prod_{t=1}^{T} y_{\pi_t}^t \quad (10)$$

O próximo passo é definir um mapeamento β a partir dos conjuntos de caminhos para o conjunto Y de possíveis rótulos. Isso é feito primeiramente removendo os rótulos repetidos e então removendo os rótulos 'vazios' dos caminhos. Por exemplo, $\beta(\phi aa\phi\phi abb) = \beta(a\phi ab\phi) = aab$. A probabilidade condicional de algum rótulo $l \in Y$ pode ser calculado pela soma das probabilidades de todos os caminhos mapeados por β.

$$p(l|x) = \sum_{\pi \in \beta^{-1}(l)} p(\pi|x) \quad (11)$$

Sendo a equação 11 considerada por [24] um problema intratável, é necessário estabelecer um modo de obter um caminho eficiente para esse cálculo. O autor então sugere um algoritmo *forward-backward* similar ao empregado no treinamento de HMM [25]. Esse conjunto de diferentes caminhos para o mesmo rótulo é o que permite à CTC usar dados não segmentados, pois remove a exigência do alinhamento da sequência de entrada e saída.

Após esse treinamento, é feito a fase da decodificação, Fig. 6, onde é possível rotular uma sequência de dados de entrada x desconhecidos escolhendo a rotulação mais provável em l^*:

$$l^* = \arg\max_l p(l|x) \quad (12)$$

O próximo passo é formular a função objetivo. Dado o conjunto de dados $S = (x, l)$ de amostras de treinamento onde x é a entrada e l é a saída desejada, o objetivo é maximizar a probabilidade da saída dado o conjunto de amostras de treinamento, o que corresponde à minimização da função objetivo, através da diferenciação da Eq. 13 com respeito as saídas da rede BLSTM.

$$O(S) = -\sum_{(x,l)\in S} \ln(P(l|x)) \quad (13)$$

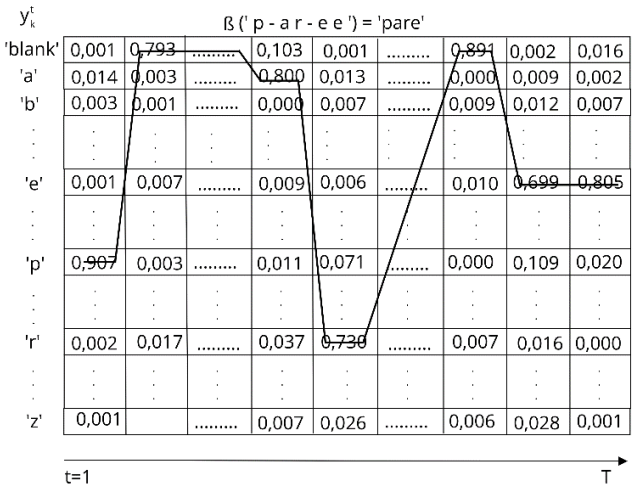

Figura 6. Exemplo da decodificação.

IV. EXPERIMENTOS E RESULTADOS

Os experimentos foram realizados com a finalidade de comparar a robustez da abordagem proposta utilizando BLSTM e CNN. A base foi dividida em 70% para treinamento e 30 % para teste, seguindo os mesmos critérios de experimentações de [19] e [14]. As abordagens foram avaliadas usando validação cruzada k-*fold*, com k = 10. Esta validação evita superposição dos conjuntos de teste. Sua estimativa é calculada como a média das estimativas de acerto em cada iteração. Para a execução desses experimentos, foi utilizada a linguagem *Python* 3.4, com o auxílio das bibliotecas *Numpy*, *Theano* e *Keras*.

O primeiro experimento, Tabela 1, consistiu no emprego de CNN e BLSTM de forma única. A rede CNN analisa o padrão do espectrograma como uma imagem, enquanto BLSTM analisa a estrutura sequencial da voz. CNN foi composta por 4 camadas convolucionais e subamostragem. Na camada de convolução foi utilizado *kernels* de dimensão 3x3 e na de subamostragem filtro de dimensão 2x2. Foram utilizados mapas de características de tamanhos: 32, 32, 64, 64 respectivamente, em cada camada de convolução. Por fim, na última etapa existe uma rede MLP que contém 512 neurônios na camada oculta e 6 neurônios na camada de saída, onde cada neurônio representa uma palavra.

TABELA I
EXPERIMENTOS UTILIZANDO CNN E BLSTM DE FORMA ÚNICA

Modelo	Acurácia (%)			
	Limpa	Fábrica	Conversa	Volvo
BLSTM	96,410 ± 5,01	45,232 ± 14,47	40,729 ± 13,91	96,410 ± 5,01
CNN	**97,314** ± 2,43	**91,751** ± 2,44	**90,972** ± 3,93	**97,313** ± 2,43

De forma semelhante, a BLSTM foi treinada diretamente com 40 bancos de filtros na escala mel, com 512 blocos de memória, utilizando uma taxa de aprendizado de 10^{-3}. Estes parâmetros foram definidos baseado em trabalhos como [12] e [26]. A Fig. 7, ilustra este processo de treinamento, obtendo como objetivo verificar o poder desses modelos de forma separada, buscando encontrar o equilíbrio entre cada uma dessas máquinas para obter melhores resultados.

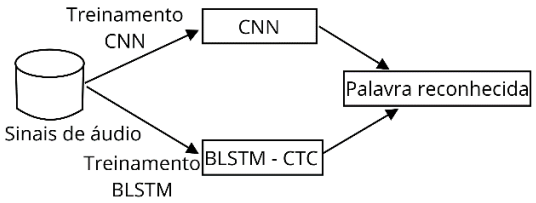

Figura 7. Processo de treinamento com CNN e BLSTM.

Para o segundo experimento, Tabela II, inicialmente foi feito a rotulação fonética, sob supervisão humana, nas bases de áudio, através de um *pluging EasyAlign* para o Praat (disponível em http://www.fon.hum.uva.nl/praat/).

A rotulação foi feita em 400 amostras da base. Para cada áudio, foram mapeados os fonemas e a duração de tempo (em milissegundos) de cada um. Nesse processo foram utilizados 15 fonemas, e mais uma classe que representa o silêncio. Em seguida, na etapa de extração de características foi gerado um vetor de 600 dimensões. Este vetor corresponde a uma janela de contexto composta por 15 *frames*, em que cada um corresponde a um vetor de 40 dimensões (Fig. 8). Cada componente de um *frame* é a energia do sinal acústico de duração de 25 *ms* em um dos 40 filtros na escala mel. O *frame* central é aquele empregado na classificação. Os sete *frames* adjacentes, à esquerda e à direita, formam o contexto. Nesta janela, os *frames* não cobrem eventos distintos, isto é, existe um entrelaçamento. Eles estão sobrepostos em intervalos de duração de 10ms. A ideia empregada neste processo de extração de características foi proposta por [26] e adotado também por [14].

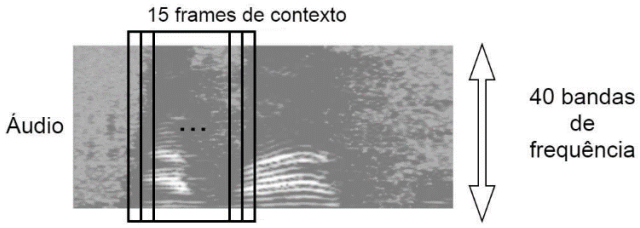

Figura 8. Extração de coeficientes.

Em seguida, a CNN foi composta por duas camadas convolucionais e subamostragem. Na camada de convolução utilizou-se *kernels* de tamanho 3x3 seguido pela camada de subamostragem com filtro de tamanho 2x2. Na primeira camada foram usados 20 mapas de características e na segunda 50. Por fim, a camada MLP possui 200 neurônios na camada oculta e 16 neurônios de saída, onde cada neurônio corresponde a um fonema, servindo como entrada para o treinamento do modelo sequencial. Para o caso HMM, para cada palavra foi criado um modelo HMM diferente, e para BLSTM, o treinamento consistiu em 256 blocos de memória, com uma taxa de aprendizado de 10^{-3}, parâmetros estes, que

foram baseados nos trabalhos de [14] e [24]. A Fig. 9 ilustra estes processos de forma genérica.

TABELA II
EXPERIMENTOS COM CNN E BLSTM UTILIZANDO ROTULAÇÃO FONÉTICA NAS BASES DE ÁUDIO

Modelo	Acurácia (%)			
	Limpa	Fábrica	Conversa	Volvo
CNN-HMM	91,679 ± 25,42	82,392 ± 23,73	77,096 ± 21,24	91,679 ± 25,42
CNN-BLSTM	**100**	**92,688** ± 6,10	**85,009** ± 4,91	**100**

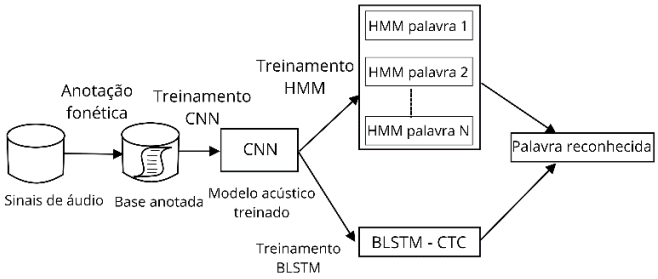

Figura 9. Processo de treinamento com HMM e BLSTM.

Para o terceiro experimento, Tabela III, foi utilizado uma abordagem com rotulação automática com 40 coeficientes de banco de filtros na escala mel e blocos dinâmicos de 50 *frames* de contexto, aprendendo a estrutura sequencial dos blocos para cada palavra. Através dessa abordagem, é eliminada a supervisão humana, facilitando o treinamento em bases com grande quantidade de amostras. As configurações da CNN, HMM e BLSTM foram as mesmas utilizadas no experimento anterior.

TABELA III
EXPERIMENTOS COM CNN E BLSTM UTILIZANDO ROTULAÇÃO AUTOMÁTICA NAS BASES DE ÁUDIO

Modelo	Acurácia (%)			
	Limpa	Fábrica	Conversa	Volvo
CNN-HMM	90,769 ± 24,74	72,329 ± 19,81	71,509 ± 18,75	89,687 ± 24,42
CNN-BLSTM	**95,531** ± 2,90	**74,327** ± 5,77	**72,992** ± 6,44	**95,531** ± 4,62

Os resultados da Tabela 1, mostra o poder da CNN em bases contaminadas com ruídos, obtendo uma acurácia de 90% em média com desvio padrão inferior a 4%. Para BLSTM, acredita-se que a baixa acurácia encontrada é dada pela pequena quantidade de amostras para o treinamento disponível na base biochaves, dispondo de pouca variação de sequências para cada palavra.

Nas Tabelas II e III, é possível verificar que a abordagem CNN-BLSTM obteve melhores resultados em ambos experimentos quando comparados a CNN-HMM.

A modelagem acústica proposta pela CNN, neste trabalho, é o principal modelo responsável pela robustez ao ruído, o papel da BLSTM assim como HMM é para modelar as sequências.

No geral, nos resultados para o ruído conversa e fábrica há um aumento nas taxas de erro comparado as outras bases, isto ocorre devido aos ruídos serem mais espalhados em diversas faixas de frequência. No espectrograma da palavra 'avance' nas Figs. 10, 11, 12 e 13 é possível perceber diferentes frequências produzidos pelos ruídos.

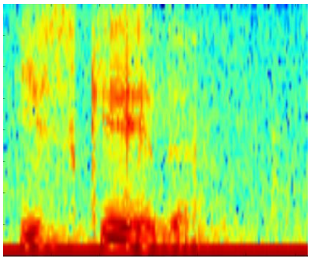

Figura 10. Espectrogramas da palavra avance na base limpa.

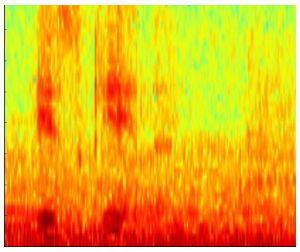

Figura 11. Espectrogramas da palavra avance na base com ruído conversa.

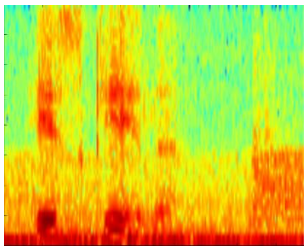

Figura 12. Espectrogramas da palavra avance na base com ruído fábrica.

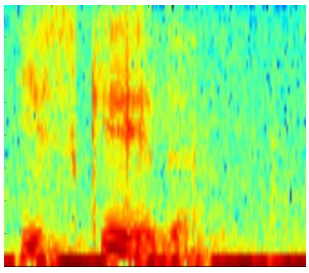

Figura 13. Espectrogramas da palavra avance na base com ruído volvo.

V. CONCLUSÃO

Neste trabalho foi avaliado o desempenho de um modelo acústico a base da CNN, rede robusta na presença de ruído, e duas abordagens diferentes para a modelagem sequencial, a HMM, máquina tradicional utilizada em sistemas ASR, e BLSTM com sua arquitetura recorrente sendo capaz de armazenar informações por um longo período.

Existem várias técnicas para a diminuição do ruído na

literatura, porém, nestes experimentos não foi aplicado nenhum tipo de pré-processamento, ou seja, os dados acústicos serviram diretamente como entrada para os modelos abordados.

Através dos experimentos pode-se constatar que a rotulação fonética foi o fator relevante para melhores taxas de acerto, em contrapartida essa abordagem exige um maior esforço pois precisa da supervisão humana, ficando inviável sua utilização em bases com grande quantidade de amostras.

Os resultados mostram que o modelo CNN-BLSTM obteve melhores resultados no cômputo geral, reforçando a capacidade de o modelo lidar com a dependência do tempo através de sua recorrência.

Como trabalho futuro, pretende-se estender essa abordagem para o reconhecimento de fala contínuo, aproximando assim, da comunicação natural entre seres humanos. Realizar também experimentos independentes de locutor, para isso, é necessária uma grande base de dados com diferentes amostras para treinamento.

AGRADECIMENTOS

Agradecemos à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pela concessão da bolsa de estudos durante o período de realização deste trabalho.

REFERÊNCIAS

[1] Seltzer, M. L., Yu D., and Wang Y. "An investigation of deep neural networks for noise robust speech recognition", IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 7398-7402, 2013.

[2] Graves, A., Mohamed, Liwicki, M., Bunke, H., Shmidhuber, J. e Fernandez, S. "Unconstrained on-line handwriting recognition with recurrent neural networks", Advances in Neural Information Processing Systems, pp. 577-584, 2008.

[3] Sutskever, I., Mohamed, Martens, J. e Hinton, G. E. "Generating text with recurrent neural networks", Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 1017-1024, 2011.

[4] Mikolov, T, Mohamed, Karafiát, M. e Burget, L., Cernocky, J. e Khudanpur, S. (2010). "Recurrent neural network based language model", Interspeech, vol. 2, pp. 3, 2010.

[5] Deng L. and Dong Y. "Deep Learning: Methods and Applications". Foundations and Trends® in Signal Processing, vol. 7, no. 3-4, pp. 197-387, 2014.

[6] Jordan, M. I. "Attractor dynamics and parallellism in a connectionist sequential machine". Lawrence Erlbaum Associates, 1986.

[7] Elman, J. L. "Finding structure in time". Cognitive science, Elsevier, vol. 14, no. 2, pp. 179-211, 1990.

[8] Gers, F. A., Schmidhuber, J., and Cummins, F. "Learning to forget: Continual prediction with lstm". Neural computation, MIT Press, vol. 12, no. 10, pp. 2451-2471, 2000.

[9] Wollmer, Martin and Zhang, Zixing and Weninger, Felix and Schuller, Bjorn and Rigoll, Gerhard. "Feature enhancement by bidirectional LSTM networks for conversational speech recognition in highly non-stationary noise", IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6822-6826, 2013.

[10] SELTZER, Michael L.; YU, Dong; WANG, Yongqiang. "An investigation of deep neural networks for noise robust speech recognition". In: IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 7398-7402, 2013.

[11] Narayanan, Arun and Wang, DeLiang, "Joint noise adaptive training for robust automatic speech recognition", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2504-2508, 2014.

[12] Graves, Alex, Navdeep Jaitly, and Abdel-rahman Mohamed. "Hybrid speech recognition with deep bidirectional LSTM." Automatic Speech Recognition and Understanding (ASRU), IEEE Workshop on. IEEE, 2013.

[13] Li, Jinyu et al. An overview of noise-robust automatic speech recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing, vo. 22, no. 4, pp. 745-777, 2014.

[14] Santos, R. M., Matos, L. N., Macedo, H. T., Montalvão, J. Speech Recognition in Noisy Environments with Convolutional Neural Networks. In: 2015 Brazilian Conference on Intelligent Systems (BRACIS). IEEE, pp. 175-179, 2015.

[15] Graves, Alex. "Sequence transduction with recurrent neural networks." arXiv preprint arXiv:1211.3711, 2012.

[16] EYBEN, Florian et al. "From speech to letters-using a novel neural network architecture for grapheme based asr". In: Automatic Speech Recognition & Understanding, ASRU. IEEE Workshop on. IEEE, pp. 376-380, 2009.

[17] GRAVES, Alex; MOHAMED, Abdel-rahman; HINTON, Geoffrey. "Speech recognition with deep recurrent neural networks". In: IEEE international conference on acoustics, speech and signal processing, pp. 6645-6649, 2013.

[18] MARTINS, Ramon Mayor; YNOGUTI, Carlos Alberto. "Normalização do locutor em sistemas de reconhecimento de fala para usuários crianças". In: Proceedings of the 13th Brazilian Symposium on Human Factors in Computing Systems. Sociedade Brasileira de Computação, pp. 381-384, 2014.

[19] RAULINO, Christiane; D., Dami; MONTALVAO, Jugurta. "Análise de espectro através da detecção de eventos acústicos elementares no plano tempo-frequência.

[20] VARGA, Andrew; STEENEKEN, Herman JM. "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems". Speech communication, vol. 12, no. 3, pp. 247-251, 1993.

[21] SONG, William; CAI, Jim. End-to-End Deep Neural Network for Automatic Speech Recognition, 2015.

[22] LECUN, Yann et al. "Gradient-based learning applied to document recognition". Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.

[23] HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. "Long short-term memory". Neural computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[24] GRAVES, Alex. Supervised sequence labelling. "Supervised Sequence Labelling with Recurrent Neural Networks". Springer Berlin Heidelberg, pp. 5-13, 2012.

[25] RABINER, Lawrence R. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, vol. 77, no. 2, pp. 257-286, 1989.

[26] ABDEL-HAMID, Ossama et al. "Convolutional neural networks for speech recognition". IEEE/ACM Transactions on audio, speech, and language processing, vol. 22, no. 10, pp. 1533-1545, 2014.

[27] GEMAN, S; BIENENSTOCK, E.; DOURSAT, R. "Neural networks and the bias/variance dilemma". MIT Press, v.4, n, 1, p. 1-58, 1992.

[28] JAITLY, N. "Exploring Deep Learning Methods for discovering features in speech signals". Tese (Doutorado) – Universidade de Toronto, 2014.

Luciana Maiara Queiroz de Santana nasceu em Aracaju/SE, Brasil em 1992. Graduou-se em Ciência da Computação pela Universidade Tiradentes (UNIT) em 2014. Atualmente é estudante de mestrado em Ciência da Computação pela Universidade Federal de Sergipe (UFS).

Rafael Meneses Santos nasceu em Frei Paulo/SE, Brasil, em 1990. Graduou-se em Sistemas de Informação pela UFS em 2014. Obteve o título de Mestre em Ciência da Computação também pela UFS em 2016, atuando na área de Computação Inteligente. Atualmente, é aluno do programa de doutorado em Ciência da Computação pela UFBA e atua como Técnico em Tecnologia da Informação na UFS.

Leonardo Nogueira Matos nasceu em Nasceu em Fortaleza/CE, Brasil, em 1969. Graduou-se em Ciência da Computação pela UFC em 1991. Obteve o título de Mestre em Matemática Aplicada pela UNICAMP em 1993 e o título de Doutor em Engenharia Elétrica pela UFCG em 2004. É professor Associado do DCOMP/UFS e membro permanente dos Programas de Pós-Graduação em Ciência da Computação

(PROCC/UFS). Junto com o professor Hendrik, é líder do Grupo de Pesquisa em Inteligência e Imagens (Pii/CNPq), atua na área de Reconhecimento de Padrões, principalmente padrões de fala e imagens.

Hendrik Teixeira Macedo nasceu em Aracaju/SE, Brasil, em 1977. Graduou-se em Ciência da Computação pela UFS em 1998. Obteve o título de Mestre em Ciência da Computação pela UFPE em 2001 e o título de Doutor em Ciência da Computação também pela UFPE em 2006, tendo realizado estágio de doutoramento "*sandwich*" pela Universidade de Paris VI no ano de 2002. De julho de 2006 até o presente momento, atua como professor efetivo Adjunto do DCOMP/UFS e como membro permanente dos Programas de Pós-Graduação em Ciência da Computação (PROCC/UFS) e Pós-Graduação em Engenharia Elétrica (PROEE/UFS). Como líder do Grupo de Pesquisa em Inteligência e Imagens (Pii/CNPqs), atua principalmente na investigação sobre interfaces naturais vocais e personalizáveis.