Curso de Ciência da Computação
Universidade Estadual de Mato Grosso do Sul

ESTUDO E ANÁLISE DE MÉTODOS PARA RECONHECIMENTO DE PALAVRAS DITAS

Raiza Artemam de Oliveira
Willian de Sousa Santos

Prof. MSc. André Chastel Lima (Orientador)

DOURADOS-MS
2016

Estudo e Análise de Métodos para Reconhecimento de Palavras Ditas

Raiza Artemam de Oliveira

Willian de Sousa Santos

Este exemplar corresponde à redação final da monografia da disciplina Projeto Final de Curso devidamente corrigida e defendida por Raiza Artemam de Oliveira e Willian de Sousa Santos e aprovada pela Banca Examinadora, como parte dos requisitos para a obtenção do título de Bacharel em Ciência da Computação.

Dourados, 16 de Novembro de 2016

Prof. MSc. André Chastel Lima

Curso de Ciência da Computação

Universidade Estadual de Mato Grosso do Sul

ESTUDO E ANÁLISE DE MÉTODOS PARA RECONHECIMENTO DE PALAVRAS DITAS

Raiza Artemam de Oliveira

Willian de Sousa Santos

Novembro de 2016

BANCA EXAMINADORA:

Prof. MSc. André Chastel Lima (Orientador)

Sistemas de Computação – UEMS

Prof. Dr. Rubens Barbosa Filho

Inteligência Artificial – UEMS

Prof[a]. Dr[a]. Glaucia Gabriel Sass

Georreferenciamento – UEMS

Computer science is no more about computers than astronomy is about telescopes, biology is about microscopes or chemistry is about beakers and test tubes. Science is not about tools, it is about how we use them and what we find out when we do..

Edgar Dijkstra

Agradecimentos

Gostariamos de agradecer ao professor André Chastel Lima pela dedicação e paciência durante o desenvolvimento deste trabalho. A professora Maria de Fátima pela ajuda no desenvolvimento do texto. Ao professor coordenador Nilton Cézar de Paula. A secretária dona Jandira pela atenção dedicada as nossas vidas acadêmicas. Aos professores presentes na banca. Por fim agradecemos a todos os professores que contribuiram nessa jornada.

Eu, Raiza, agradeço aos meus pais, Eneias e Sandra, por todo o apoio. Aos meus tios, Marcia e Juraci, por serem sempre prestativos. Aos meus avós Maria Lucilene (*in memorian*) e João. Por fim agradeço a professora Adriana Betania de Paula Molgora.

Eu, Willian, agradeço ao meu pai, João e minha mãe, Ana Lúcia, por terem me dado o incentivo e um ambiente propício aos estudos.

RESUMO

O reconhecimento de fala é um campo de estudo muito amplo e com diversas etapas envolvidas. O processo de reconhecimento de uma palavra isolada se inicia na captação da onda sonora, passa por vários métodos para tratamento desta onda e extração de descritores do sinal. Uma vez obtido estes descritores se inicia a fase de comparação. Nesta fase, um algoritmo deve ser aplicado para o reconhecimento do sinal de entrada. Existem vários algoritmos que utilizam técnicas diferentes. Para chegar a uma solução no reconhecimento, além dos algoritmos e descritores, o ambiente, o contexto da aplicação e os recursos disponíveis também devem ser considerados na decisão de quais técnicas devem ser empregadas. Neste trabalho trazemos uma introdução a estas técnicas. São estudados métodos desde a primeira etapa que consiste na captura e processamento do sinal digital até a fase final, onde são considerados os algoritmos e técnicas aplicados ao reconhecimento de uma palavra isolada.

Palavras-chave: MFCC. SOM. HMM. ALSA. Reconhecimento de Fala.

ABSTRACT

Speech recognition is a very broad field with study with several steps involved. The process of recognition of an isolated word starts the capture of the signal it applied to several methods for treatment with the goal of feature extracting and extracting features. Once obtained these features it starts the comparison phase. At this stage, an algorithm must be applied to the recognition of the input signal. To obtain a solution to recognition algorithms and features extraction, the environment, the application context and the resources available should also be considered in deciding which techniques should be employed. In this paper we shows an introduction to these techniques. The methods are studied from the begining of the capture and digital signal processing to the final stage, which are considered the algorithms and techniques applied to the recognition of a isolated word.

Keywords: MFCC. HMM. Alsa. Speech Recognition.

SUMÁRIO

Lista de siglas

ALSA - Advanced Linux Sound Architeture

API - Application Programming Interface

CDMA - Code Division Multiple Access

DBNF - Deep Bottle-Neck Features

DCT - Discrete Cosine Transform

FDP - Função Densidade de Probabilidade

FFT - Fast Fourier Transform

GSM - Groupe Special Mobile

HMM - Hiden Markov Model

LPC - Linear Prediction Coefficients

MFCC - Mel Frequency Cepstral Coefficients

PCM - Pulse Code Modulation

PLP - Perceptual Linear Prediction

PNCC - Power-Normalized Cepstral Coefficients

RASTA-PLP - RelAtive SpecTrAl Perceptual Linear Prediction

RIFF - Resource Interchange File Format

WAVE - Waveform Audio File Format

Lista de tabelas

Lista de ilustrações

1 INTRODUÇÃO

Nos primeiros sistemas computacionais a comunicação entre pessoas e máquinas era realizada através de terminais por linha de comando ou cartões perfurados. Apenas especialistas conseguiam utilizar estes sistemas. Depois, no início da década de 70, com a criação do mouse e a introdução da interface gráfica os sistemas tornaram-se mais amigáveis ao usuário, podendo ser utilizados por pessoas comuns sem necessidade de conhecimento técnico. Com o passar dos anos a interação entre pessoas e máquinas tornou-se mais intuitiva com às diversas interfaces entre o usuário e o sistema. No fim da década de 70 iniciaram-se as pesquisas de reconhecimento de fala. Interfaces por meio de fala são utilizadas em diversas áreas, tais como: sistemas embarcados, automação residencial, operações bancárias, conversão fala texto e dispositivos móveis.

O reconhecimento da fala é um campo de estudo amplo e necessário às diversas tecnologias que a utilizam como um meio de comunicação entre o usuário e o sistema. Utilizar a fala como entrada de um sistema torna a comunicação entre o usuário e o sistema mais direta, intuitiva, rápida e precisa. Como um campo de ampla aplicacação, o reconhecimento de fala tem diversos projetos em diferentes partes do mundo. Dentre os quais se destaca o projeto CMU Sphinx da universidade americana Carnegie Mellon. O projeto já tem cerca de 20 anos de pesquisas na área de reconhecimento de fala e de voz. Trata-se de um projeto *open source* voltado para linux, mas também conta com uma versão em java multiplataforma. O CMU Sphinx oferece suporte para várias linguagens, dentre elas o inglês, alemão, russo, francês e espanhol. O reconhecimento de fala pode ser classificado de acordo com o tamanho do vocábulario, de acordo com os algoritmos utilizados e de acordo com o tipo de fala a ser reconhecida (contínua ou discreta).

1.1 Justificativa

O reconhecimento de fala é campo de estudo muito amplo e com diversas aplicações, como já foi citado acima. Com base nisso propomos realizar um estudo sobre o reconhecimento de palavras isoladas em um fluxo contínuo, ou seja, captar palavras chaves em meio a um diálogo, para análisar os métodos que melhor se encaixam para a vigilância de ambientes, com baixo custo computacional para a inclusão de sistemas embarcados.

1.2 Objetivos

O objetivo deste trabalho é realizar estudos preliminares de métodos de reconhecimento de palavras isoladas em fluxo contínuo independente do locutor. Analisar os algoritmos utilizados, suas vantagens e desvantagens. Apresentar os resultados para um pequeno vocábulario. Analisar os métodos estudados afim de implementação em um sistema embarcado.

1.2.1 Objetivo específico

Apontar a melhor solução para reconhecimento de palavras ditas independente de locutor em fluxo contínuo com baixo custo computacional.

1.3 Metodologia

A metodologia adotada para a realização deste trabalho consistiu em pesquisa em livros, sites, artigos e notas de aula sobre o tema abordado e seus diversos aspectos, após a fase de pesquisa foi realizado o estudo de algoritmos aplicados ao reconhecimento de fala, foram estudados os algoritmos HMM, SOM e um método de correlação entre características da fala, estes três métodos foram implementados, a explicação sobre a implementação é realizada no capítulo 6, após a implementação foram realizados testes preliminares e avaliação dos algoritmos, por último foi feita a documentação do trabalho explicando os principais conceitos no campo de reconhecimento de fala e processamento de sinais aplicados a esta área.

1.4 Organização do Trabalho

No capítulo 2 é feita uma explicação do que é relevante para este trabalho com base na literatura. No capítulo 3 é feita uma explicação de como é realizada a captação de áudio. No capítulo 4 a transformação do áudio em coeficientes mel-cepstrais. No capítulo 5 é realizada a explicação sobre as técnicas estudadas neste trabalho.No capítulo 6 é feita uma breve explicação sobre a implementação dos métodos. No capítulo 7 são expostos os resultados obtidos. O capítulo 8 traz as considerações finais sobre o trabalho desenvolvido e sugestão para futuros trabalhos.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo traz uma revisão bibliográfica dos sistemas de reconhecimento de fala e processamento de sinais digitais, uma vez que estes são os assuntos que constituem a base deste trabalho.

2.1 Sistemas de reconhecimento de fala

Sistemas de reconhecimento de fala permitem que computadores equipados com microfone interpretem a fala. De acordo com Rabiner e Juang (1993), os sistemas de reconhecimento de fala podem ser classificados em três grupos de acordo com a técnica utilizada. Estes grupos são reconhecedores por inteligência artificial, reconhecedores por comparação de padrões, reconhecedores baseados na análise acústico-fonética.

2.1.1 Reconhecedores baseados em inteligência artificial

Os sistemas de reconhecimento de fala que utilizam a inteligência artificial usam propriedades tanto dos reconhecedores por comparação de padrões quanto dos reconhecedores baseados na análise acústico-fonética. Sistemas com redes neurais são encaixados nesta classe. As redes Multilayer Perceptron usam uma matriz de ponderação que representa as conexões entre os nós da rede, e cada saída está associada a uma unidade a ser reconhecida (MORGAN; SCOFIELD, 1991).

A abordagem de inteligência artificial baseia-se no processo humano natural de ouvir, analisar e tomar uma decisão sobre as características acústicas medidas para reconher a fala. Segundo Rabiner e Juang (1993) Faz parte do processo de reconheciemento de fala pela abordagem de inteligência artificial o processo de segmentação e rotulagem usado na análise acústico-fonética. Esta abordagem aplica o conceito de que o conhecimento é dinâmico e os modelos devem adaptar-se frequentemente.

2.1.2 Reconhecedores por comparação de padrões

Estes reconhecedores usam o príncipio de que o sistema foi treinado para reconhecer os padrões. Os sistemas por reconhecimento de padrões possuem duas fases diferentes :

- ❑ Treinamento;

❑ Reconhecimento.

Durante a fase de treinamento são criados padrões de referência para o sitema. Na fase de reconhecimento compara-se os padrões obtidos com os padrões de referência criados na fase anterior e calcula-se uma medida de similaridade entre os padões. O padrão mais similar ao desconheido é escolhido como reconhecido. Os sistemas que se baseiam nos Modelos Ocultos de Markov (HMM) se encaixam nesta categoria.

Dentre as diversas razões para usar a abordagem de comparação de padrões para reconheimento de fala pode-se citar a simplicidade de uso, por ser um método de fácil entendimento que possui uma rica fundamentação matemática e é amplamente utilizado, e a robustez, trata-se de um método robusto e invariante para diferentes vocábularios, algoritmos de comparação de padrão e regras de decisão. Isto torna esta abordagem apropriada para uma vasta gama de unidades de fala, como fonemas, palavras isoladas ou frases (RABINER; JUANG, 1993).

2.1.3 Reconhecedores baseados na análise acústico-fonética

Os sistemas baseados na análise acústico-fonética decodificam o sinal de fala baseados nas características acústicas deste sinal e na relação entre elas (INCER, 1992). Os sistemas de análise desta classe devem considerar propriedades acústicas invariantes. Entre estas características estão a classificação entre sonoro e não sonoro, segmentação do sinal da fala, detecção das características que descrevem as unidades fonéticas e escolha do padrão que mais corresponde à sequência de unidades fonéticas.

Os reconhecedores baseados na análise acústico-fonética trabalham em duas etapas. O primeiro passo na análise acústico fonética é chamado de fase de segmentação e rotulagem (RABINER; JUANG, 1993). Este passo envolve a segmentação do sinal da fala em regiões discretas, no tempo, onde as propriedades acústicas do sinal são representadas por um único fonema, ou estado. Em seguida uma ou mais etiqueta fonética é associada a cada região segmentada de acordo com as propriedades acústicas. O segundo passo para o reconhecimento tenta determinar uma palavra válida a partir da sequência de etiquetas fonéticas obtidas na fase anterior. As palavras são obtidas a partir de um determinado vocabulário, as palavras obtidas fazem sentido sintático e tem significado semântico.

2.2 Processamento digital de sinais

De acordo com Ortigueira (2005) um sinal é qualquer função associada a um fenômeno físico, econômico ou social e que transporta algum tipo de informação sobre ele. Pode ser definido como uma descrição quantitativa de um dado fenômeno. A voz é um exemplo de sinal.

Os sinais podem ser classificados de diferentes formas de acordo com suas características e com o tipo de domínio e contradomínio. Segundo Ortigueira (2005) esta classificação pode ser feita de acordo com as seguintes características:

1. Variável independente: o sinal é contínuo se $t \in \mathbb{R}$ e discreto se $t \in \mathbb{Q}$. Os pontos $t_n, n \in \mathbb{Z}$ são chamados de instantes de amostragem. Sinal amostrado é o sinal discreto obtido por amostragem de um sinal contínuo.

2. Amplitude: os sinais podem ser classificados de acordo com a amplitude em :

   - ❏ Analógicos: sinal contínuo cuja amplitude pode assumir uma gama contínua de valores;
   - ❏ Quantificados: sinal cuja amplitude pode assumir, apenas, uma gama finita de valores;
   - ❏ Digitais: sinal resultante da codificação de um sinal amostrado e quantificado. A codificação consiste em atribuir a cada valor obtido por amostragem e quantificação um código.

3. Duração: os sinais cujo dominio é limitado dizem-se de duração finita, os restantes são de duração infnita.Os sinais de duração finita tambem são chamados de janela.

4. Reprodutibilade: um sinal é dito determinístico se repetindo a mesma experiência obtém-se o mesmo resultado, caso isso não seja possível então trata-se de um sinal aleatório.

5. Periodicidade: os sinais determinísticos classificam-se ainda em aperiódicos e periódicos. Os sinais aperiódicos não são repetitivos. Os sinais periódicos são repetitivos e possuem a relação $x(t) = x(t \mp T) \quad \forall \quad t$, onde T é o período. Quando $T < 2\pi$ a involvente final do sinal periódico $x(t)$ não coincide com a extensão periódica do sinal base $x_b(t)$ ocorre o fenômeno chamado *aliasing*. O fenômeno de aliasing é importante na conversão discreto-contínua e verifica-se no domínio da frequência.

6. Morfologia: formas simétricas a um eixo ou outro. Os sinais pares são simétricos ao eixo das ordenadas. Os sinais impares são simétricos ao eixo das abscissas.

7. Carater : outras medidas são consideradas. Um sinal pode ter carater escalar, vetorial. Por exemplo, o sinal de saída de um conjunto de sensores é um sinal sensorial.

A análise frequencial moderna é um conjunto de técnicas matemáticas e ou físicas que permite obter o conteúdo frequêncial de qualquer sinal, a que se chama de **espectro** . O processo de obtenção de espectro chama-se **análise espectral**. O processo numérico usado para determinar o espectro é chamado de **estimação espectral**. A estimação espectral é feita em sinais de fonte fisíca, como a voz, durante um intervalo de tempo finito. Na prática o conteúdo frequencial de um dado sinal não é uniforme. Assume valores significativos em intervalos chamados bandas. A Tabela 1, baseada em Ortigueira (2005), mostra alguns sinais e suas bandas.

Tabela 1: Bandas ocupadas por alguns sinais

Sinal	de	a
Eletrocardiograma	0 Hz	150 Hz
Eletroencefalograma	0 Hz	100 Hz
Voz	100 Hz	4000 Hz
Ruído do vento	100 Hz	1000 Hz
Ruído de tremor de terra	0.01 Hz	10 Hz
Radiodifusão	0.03 MHz	3 MHz
Onda curta	3 GHz	30 GHz
Radar, satélite, comun. espaciais	300 GHz	300 THz
Luz visível	370 THz	770 THz

A designação de filtro habitualmente usada em referência aos sistemas lineares, deriva da possibilidade de certos sistemas eliminarem ou atenuarem fortemente certas bandas.

3 CAPTURA DE ÁUDIO

Este capítulo traz uma discussão sobre a captura do sinal sonoro, as bibliotecas usadas para capturar a onda e os formatos para armazenar o áudio.

A captura do sinal de áudio é uma parte fundamental para o desenvolvimento de um sistema reconhecedor de fala. Existem bases de dados disponíveis para testes em que a captura do sinal de áudio não é necessária, um vez que estas bases disponibilizam os arquivos de áudio. Um exemplo de base de dados de voz é a Aurora-1, esta base é construída por sinais de fala limpos e degradados através de oito tipos de ruídos. Neste trabalho optamos por realizar a captura do áudio pois este também faz parte do objetivo.

O som se propaga no ambiente por meio de ondas de forma contínua no tempo e no espaço a uma velocidade média de *340 metros/segundo* fazendo o ar vibrar. Esta onda sonora é capturada por meio de um microfone como uma onda analógica e é convertida para um sinal digital. A onda capturada é normalizada através de um filtro de passa-baixas. Circuitos que realizam esta conversão de onda são chamados de ADC (*analog digital converter*). O tamanho das amostras, expressa em bits, é um dos fatores que determina a precisão com que o som é representado em forma digital. Outro fator importante que afeta a qualidade de som é a taxa de amostragem. De acordo com Proakis e Manolakis (1996), o teorema de Nyquist afirma que a frequência mais elevada que pode ser representada com precisão é, no máximo, metade da taxa de amostragem.

3.1 Bibliotecas para Captura de Áudio

Para o processo de reconhecimento de fala de qualquer tipo, primeiro é necessário capturar o sinal de áudio. A fase de captura de áudio é essencial para o bom desempenho do projeto. Existem diversas bibliotecas *open-source* que oferecem funções que realizam a captura e gravação de áudio, entre elas a Allegro e OpenGL, entretanto a aplicação dessas bibliotecas implica em um maior custo computacional, uma vez que estas trazem milhares de linhas de código junto com outras funções além das necessárias para a implementação deste projeto. Com base nisso, buscou-se uma alternativa que integrasse eficiência e baixo custo computacional para aplicações em áudio.

3.1.1 ALSA

ALSA (*advanced linux sound architeture*) consiste de um conjunto de *drivers* do kernel, uma biblioteca, uma API e programas utilitários para o suporte de som no linux. Jaroslav Kysela iniciou o projeto ALSA porque os *drivers* de som do kernel Linux não estavam sendo devidamente mantidos e atualizados. Após a iniciativa, mais desenvolvedores aderiram ao projeto e a estrutura da API foi refinada. ALSA foi incorporada ao kernel oficial do Linux 2.5. A biblioteca fornecida pelo ALSA, libasound, fornece uma nomeação lógica dos dispositivos de hardware. Os nomes podem ser de dispositivos de hardware reais ou plugins (TRANTER, 2004). Os dispositivos de hardware usam o formato $HW : i, j$, onde i é o número do cartão e j do dispositivo do cartão. Uma placa de som tem um buffer de hardware que armazena amostras gravadas. Quando este buffer enche, ele gera uma interrupção. O driver de som do kernel, em seguida, utiliza o acesso direto à memória para transferir as amostras para um *buffer* de aplicativo na memória. O tamanho deste buffer pode ser programado por chamadas da biblioteca ALSA. Caso o buffer seja muito grande a transferência geraria uma latência excessiva. ALSA resolve isso dividindo o *buffer* em fragmentos e transfere os dados fragmentados. A Figura 1 ilustra a repartição do *buffer* em períodos, quadros e amostras, onde:

- ❏ Períodos: contém fragmentos de dados em um ponto no tempo.
- ❏ Fragmentos: Menor unidade de um periodo.
- ❏ Amostra: valores(nesse caso, contém 2 bytes, 1 é o Menor Bit Significativo e o outro Maior Bit Significativo).

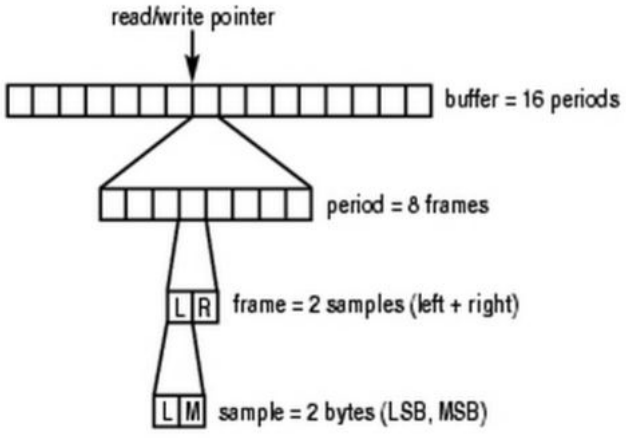

Figura 1: Buffer de aplicação. *fonte:(TRANTER, 2004)*

De acordo com Tranter (2004) a API ALSA oferece seis principais interfaces. São elas a interface de controle, interface MIDI raw, interface de tempo, interface de sequência, interface mixer e interface de PCM. Esta última, gerencia a captura e reprodução de áudio digital.

3.2 Arquivos WAVE

O formato de áudio adotado foi o WAVE. Neste tipo de formato o som é armazenado em sequências numéricas. O áudio é convertido em dados e armazenado bit a bit. O WAVE (.wav) foi criado pela IBM e pela Microsoft, nos anos oitenta e tem suporte a uma série de resoluções de bit, taxas de amostragens e canais de áudio. A taxa de amostragem em arquivo .wav refere-se ao número de amostras por segundo. O CD possui uma taxa de amostragem de $44,100$, o que significa que cada segundo de áudio tem $44,100$ amostras. A quantidade de bits usada determina quanta informação pode ser armazenada no arquivo. A quantidade de bits também interfere na amplitude do sinal. Em uma gravação de 8 bits estará disponível 256 níveis de amplitude, variando de 0 à 255. Em uma gravação de 16 bits a quantidade de níveis de amplitude disponíveis passa a $65,536$, variando entre $-32,768$ até 32767. A quantidade de 16 bits é suficiente para este projeto.

3.2.1 Cabeçalho WAVE

O cabeçalho de um arquivo .wav possui 44 bytes e é organizado como mostrado na Tabela 2.

Tabela 2: Formato de um cabeçalho de arquivo wave

Posição	Valor	Descrição
1 - 4	RIFF	Define como um arquivo RIFF
5 - 8	Tamanho do arquivo (int)	Tamanho máximo do arquivo em bytes
9 - 12	"WAVE"	Arquivo tipo cabeçalho wave
13 - 16	"fmt"	Marca formato chunk
17 - 20	16	Tamanho do formato dos dados
21 - 22	1	Formato tipo PCM
23 - 24	2	Quantidade de canais
25 - 28	44100	Taxa de amostragem
29 - 32	176400	(taxa de amostragem * bits por amostra * canais) / 8
33 - 34	4	limites
35 - 36	16	Quantidade de bits por amostra
37 - 40	data	Marca o início da seção de dados
41 - 44	Tamanho do arquivo (dados)	Tamanho da seção de dados

4 PRÉ-PROCESSAMENTO

Neste capítulo discutimos as técnicas usadas para o pré-processamento do sinal de áudio. O processo para reconhecimento de fala pode ser divido em várias etapas. O sinal de áudio é recebido do meio externo através de um transdutor e convertido para um sinal digital, a partir deste momento devemos tratar este sinal. A Figura 2 ilustra as etapas do processo de extração de características *mel-cepstrais*, também chamadas características **MFCC** (*mel frequency cepstral coefficients*).

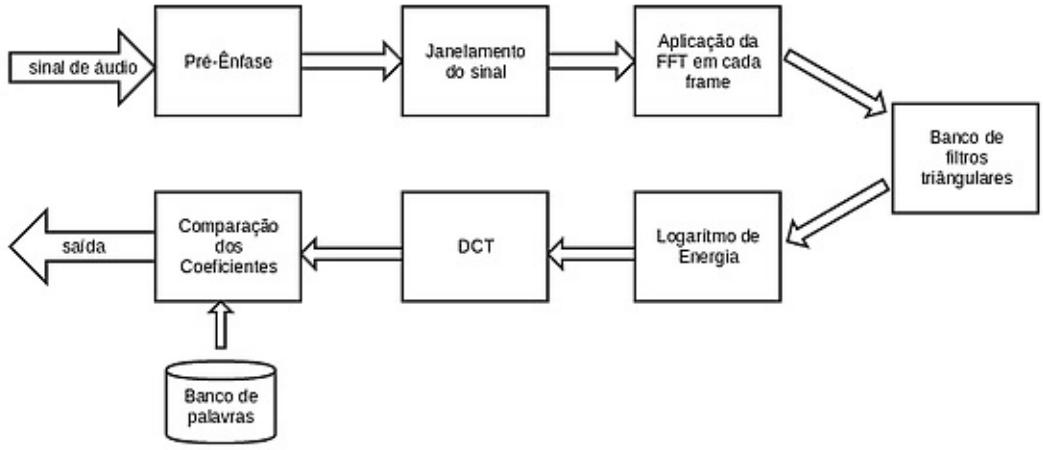

Figura 2: Etapas para extração de coeficientes MFCC.

O sinal recebido deve passar pelo pré-processamento para reduzir as interferências externas do sinal e ressaltar as informações úteis. Durante a etapa de pré-ênfase o sinal é normalizado. A normalização da amplitude do sinal garante que sons em diferentes alturas sejam processados igualmente. Os períodos de silêncio do sinal são retirados para que apenas dados importantes sejam armazenados.

Após a etapa de pré-ênfase é realizado o janelamento do sinal, ou seja, o sinal é dividido em frames. É aplicada uma janela de Hamming para atenuar as descontinuidades causadas no início e final de cada frame. A próxima etapa é a aplicação da Transformada Rápida de Fourier (FFT - do inglês *fast fourier transform*) no sinal para obter a potência espectral.

A FFT transforma um sinal do domínio do tempo para um do domínio da frequência. A Transformada Discreta de Fourier (DFT - do inglês *discret fourier transform*) possui complexidade $O(n^2)$ e a FFT possui complexidade $O(nlogn)$, por este motivo a FFT é usada em aplicações computacionais.

4.1 Filtros Digitais

Filtros digitais são usados para separar os sinais. Para o processamento de áudio são aplicados os filtros no domínio da frequência, estes filtros selecionam certas regiões no espectro, bloqueando as demais (ORTIGUEIRA, 2005). Aplicar diferentes filtros para o sinal de voz implica em diferentes técnicas. Em (VIANA, 2013) e em (GORDILLO, 2013) é possível encontrar uma descrição mais detalhada para descritores de voz. Neste trabalho foram utilizados os filtros triângulares que estão diretamente realcionados a escala *mel*, esta é explicada na seção seguinte.

4.1.1 Escala Mel

Em 1937 Stanley Smithy Stevens, John Volkman e Edwin Newmann propuseram o uso de uma variável psicoacústica chamada **pitch** para a criação de uma escala musical perceptual de tons em intervalos igualmente espaçados, chamada escala ***mel***. A frequência ouvida pelo sistema auditivo humano é subjetiva e varia de acordo com cada indivíduo. Esta impressão subjetiva de frequência é a sensação subjetiva da intensidade ou a amplitude de um som. A escala *mel* é uma escala de pitches julgados pelos ouvintes como sendo igual em distância um do outro. O ponto de referência entre esta escala e a medição de freqüência normal é definida igualando um tom de 1000 Hz , 40 dB acima do limiar do ouvinte , com um pitch de 1000 *mels*. Abaixo de cerca de 500 Hz as escalas de *mel* e Hertz coincidem, acima disso intervalos cada vez maiores são julgados por ouvintes para produzir iteração igual aos pitches. A escala *mel* é baseada em um mapeamento entre a frequência real e o pitch aparentemente percebido do sistema auditivo humano. Para converter uma frequência em escala *mel* aplica-se a equação 1, onde f é frequência.

$$M(f) = 1125ln(1 + \frac{f}{700}) \tag{1}$$

4.1.2 Filtros Triângulares

A percepção humana de algumas frequências de sons complexos não podem ser individualmente dentro de certas bandas, quando uma dessas componentes cai fora da banda, chamada de banda crítica, ela pode ser identificada (VIANA, 2013). Isto ocorre porque a percepção de uma frequência particular pelo sistema auditivo, por exemplo f_0, é influenciada pela energia da banda crítica das frequências em torno de f_0. O valor dessa banda varia nominalmente de 10% a 20% da frequência central do som, começando em torno de $100Hz$ para frequências abaixo de $1kHz$ e aumentando em escala logarítmica acima disso. Com base

nestes fenômenos utiliza-se o logarítmo da energia total das bandas críticas em torno das frequências mel. A aproximação utilizada para este cálculo é a utilização de um banco de filtros espaçados uniformemente na escala mel, o banco de filtros triangulares.

A Figura 3 mostra um banco de filtros usados na técnica MFCC. Cada filtro calcula a média do espectro em torno de um espectro central. Quanto maior a frequência, maior é a largura da banda.

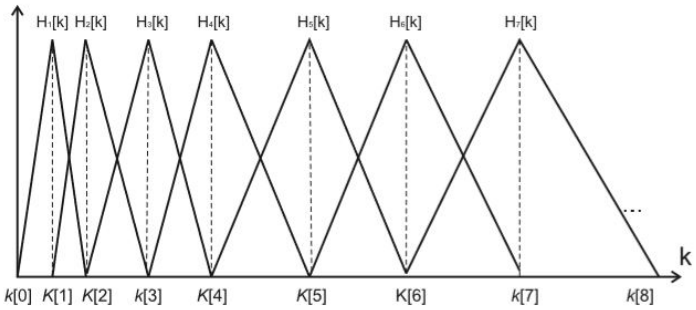

Figura 3: Banco de filtros triângulares MFCC. *fonte: (GORDILLO, 2013)*

Para determinar matematicamente os segmentos, parte-se da frequência extremas f_l e f_h que são as frequências de corte do banco de filtros em Hz. Esses valores são usados para dividir o intervalo em $B+1$ partes iguais, $k[m]$ são as frequências digitais e Mel^{-1} determina a largura do banco de filtros e é dado por

$$Mel^{-1}(m) = 700(e^{\frac{m}{1125}} - 1) \tag{2}$$

Em seguida, obtém-se a log-energia da saída de cada um dos filtros *mel*. Por fim os coeficientes MFCC são obtidos aplicando a Transformada Discreta de Cosseno (DCT - do inglês *discret cosine transform*) ao logarítmo dos coeficientes de energia obtidos no passo anterior.

5 TÉCNICAS ESTUDADAS

5.1 Modelos Ocultos de Markov

O algoritmo HMM - Hiden Markov Models (modelos ocultos de markov) - são utilizados para reconhecimento de padrões temporais como a fala, os gestos, a escrita e a bioinformática.

De acordo com Rabiner e Juang (1993) um modelo de Markov pode ser definido como um conjunto finito de estados ligados entre si por transições, formando uma máquina de estados. Estas transições estão ligadas por um processo estocástico . Há ainda um outro processo estocástico associado a um modelo de Markov, que envolve as observações de saída de cada estado. Se somente as observações de saída forem visíveis a um observador externo ao processo, diz-se então que os estados estão ocultos.

Um HMM é caracterizado por:

- ❏ Um conjunto de estados $S = \{S_1, S_2, \ldots, S_{n-1}, S_n\}$, onde n é o número de estados;
- ❏ Função de probabilidade de estado inicial $\pi = \{\pi_i\}$.

$$\pi_i = P[q_1 = S_i] \quad 1 \leq i \leq n \tag{3}$$

onde q_1 é o estado inicial $(t = 1)$.

- ❏ Função de probabilidade de transição A;
- ❏ Função de probabilidade de símbolos de saída B.

Considerando exclusivamente processos em que as probabilidades de transição não dependem do tempo e os HMMs são de primeira ordem, um HMM é considerado de primeira ordem quando a trasição do estado depende apenas da probabilidade do estado anterior mais recente. O conjunto de probabilidades de transição A é definido por:

$$A = \{a_{ij}\} \tag{4}$$

$$a_{ij} = P[q_{t-1} = S_i][q_t = S_j] \quad 1 \leq i, j \leq n \tag{5}$$

onde a_{ij} é a probabilidade de ocorrer uma transição do estado S_i para o estado S_j.
Os coeficientes a_{ij} devem obedecer às seguintes regras:

$$a_{ij} \geq 0 \quad 1 \leq i, j \leq n \tag{6}$$

$$\sum_{j=1}^{n} a_{ij} = 1 \quad 1 \leq i \leq n \tag{7}$$

A probabilidade de estar no estado S_j no instante de tempo t depende somente do instante de tempo t_j.

5.1.1 HMM e a função densidade de probabilidade

Um HMM também pode ser classificado de acordo com a função densidade de probabilidade em HMM discreto, contínuo e semicontínuo.

5.1.2 Função densidade de probabilidade

Para ser uma FDP, dada uma variável aleatória X, dizemos que $f(x)$ é uma função densidade de probabilidade de X, se e somente se $f(x)$ atender as seguintes condições:

$$f(x) \geq 0 \qquad a < x < b$$

Uma variável aleatória é uma função cujo valor é um número real determinado por cada elemento em um espaço amostral.

$$\int_a^b f(x)dx = 1 \tag{8}$$

5.1.2.1 HMM Discreto

O número de possíveis símbolos de saída é finito (RABINER; JUANG, 1993). A probabilidade de emitir o símbolo V_k no estado S_i é dada por $b_i(k)$. As propriedades da função de probabilidade B são:

$$b_i(k) \geq 0 \qquad 1 \leq i \leq n \qquad 1 \leq k \leq K$$

$$\sum_{k=1}^{K} b_i(k) = 1 \qquad 1 \leq i \leq n \tag{9}$$

As observações são discretas por natureza ou discretizadas através de uma técnica de quantização vetorial, gerando assim *codebooks* (RABINER, 1989).

5.1.2.2 HMM Contínuo

A função densidade de probabilidade é contínua. Geralmente uma função densidade elipticamente simétrica, tal como a função densidade de probabilidade Gaussiana (RABINER; JUANG, 1993). As observações são contínuas e a FDP contínua é usualmente modelada como uma mistura finita de matrizes gaussianas multidimensionais.

5.1.2.3 HMM Semicontínuo

O modelo é um caso intermediário entre contínuo e o discreto. O conjunto função densidade probabilidade é o mesmo usado para todos os estados e todos os modelos. A probabilidade de emissão dos símbolos de saída é dada por :

$$b_j(O_t) = \sum_{V_k \in \eta(O_t)}^{-} c_j(k) f(O_t|V_k) \qquad 1 \leq j \leq n \tag{10}$$

onde:

O_t é o vetor de entrada

$\eta(O_t)$ é o conjunto das funções densidade de probabilidade que apresentam os M maiores valores de $f(O_t|V_k)$, $1 \leq M \leq K$

K é o número de funções densidade de probabilidade, ou seja, os símbolos de saída

V_k é o k-ésimo símbolo de saída

$c_j(k)$ é a probabilidade de emissão do símbolo V_k no estado S_j

$f(O_t|V_k)$ é o valor da k-ésima função densidade de probabilidade.

5.1.3 Topologia

Outra maneira de classificar um HMM é de acordo com a estrutura de transição da matriz A da cadeia de markov. Existem vários modelos de HMM. A Figura 4 ilustra os principais modelos de acordo com Rabiner e Juang (1993), o ergódico totalmente conectado onde qualquer estado pode ser alcançado com um único passo, o modelo de caminhos paralelos e o modelo "left-right", também chamado de modelo Bakis.

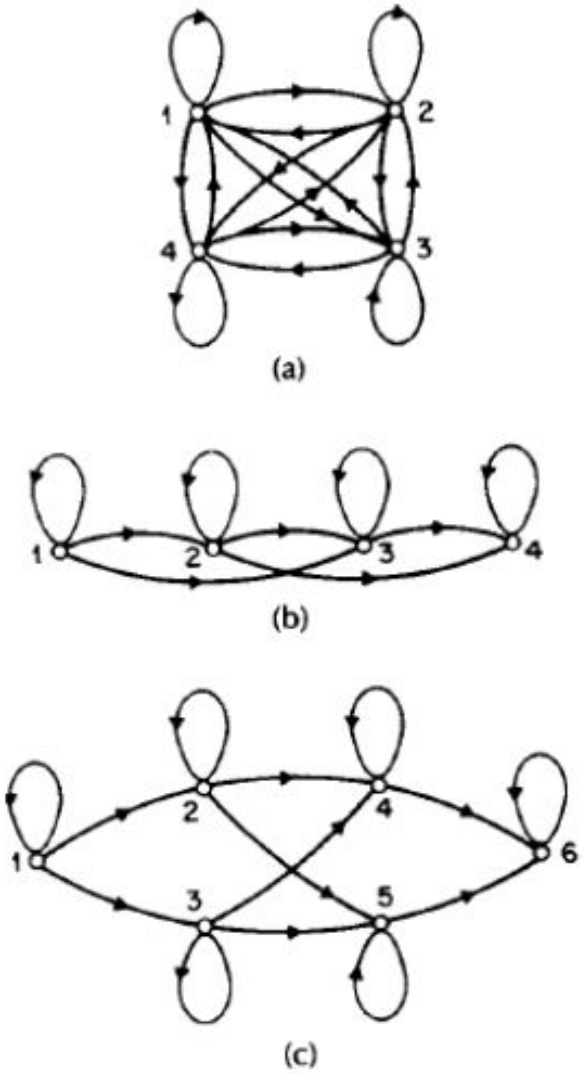

Figura 4: Exemplo de topologias de HMM. a) Modelo ergódico b) Modelo esquerda-direita c) Modelo esquerda-direita paralelo. *(RABINER; JUANG, 1993)*

A Figura 4 (a) mostra um modelo ergódico, ou totalmente conectado, neste modelo qualquer estado pode ser alcançado em um único passo a partir de qualquer estado. Segundo Rabiner e Juang (1993) as propriedades do sinal de fala são melhores modeladas em modelos *esquerda-direita* como o da Figura 4 (b) porque a sequência de estado associada ao modelo tem a propriedade de que, à medida que o tempo aumenta, o índice de estado aumenta, ou seja, os estados de desenvolvem da esquerda para a direita. O modelo ergódico gera uma matriz de transição completa, enquanto o modelo esquerda-direita gera uma matriz triangular superior.

5.1.4 Os problemas a serem resolvidos

O HMM possui três problemas básicos, que são:

1. **Problema de avaliação:** Dada a sequência de observação $O = (o_1, o_2, o_3, ..., o_n)$ e o

modelo $\lambda = (A, B, \pi)$, como calcular eficientemente $P(o|\lambda)$.

2. **Problema da busca da melhor sequência de estados.**

3. **Problema de treinamento:** como ajustar os parâmetros do modelo $\lambda(A, B, \pi)$ para maximizar $P(o|\lambda)$.

O problema da avaliação pode ser solucionado através do procedimento *Forward-Backward*. O segundo problema é solucionado com a aplicação do algoritmo de *Viterbi* e o terceiro e último problema pode ser otimizado aplicando um procedimento iterativo como o método de *Baum-Welch*. Nas seções 5.1.4.1, 5.1.4.2 e 5.1.4.3 faz-se uma explicação sobre os procedimentos para a solução dos problemas 1, 2 e 3 respectivamente.

5.1.4.1 Foward-Backward

Com a resolução do problema 1 podemos responder à algumas perguntas, se dado um modelo e uma sequência de observações, como podemos saber de que a sequência observada foi produzido pelo modelo ou, podemos ver essa solução de outra forma, um modelo é satisfatório para determinada entrada de observações

❏ Inicialização:

$$\alpha_1(i) = \pi_i b_i(O_1), \qquad 1 \leq i \leq N \tag{11}$$

❏ Indução:

$$\alpha_t + 1(j) = \sum_{i=1}^{N} \Big[\alpha_t(i) a_{ij}\Big] b_j(O_t + 1), \qquad 2 \leq t \leq T$$

❏ Término:

$$P(O|\lambda) = \sum_{i=1}^{N} \Big[\alpha_t(i)\Big]$$

5.1.4.2 Viterbi

O algoritmo de Viterbi é um algoritmo de programação dinâmica usado para encontrar a sequência de estados ocultos ótima. Dado uma sequência de estados ocultos de um HMM, o algoritmo de viterbi calcula a melhor sequência de estados baseados nas probabilidades de transição. Este algoritmo foi proposto em 1967 por Andrew Viterbi para a decodificação de códigos convolucionais em links de comunicação ruidosos. O algoritmo também possui

aplicações em redes CDMA e GSM, modem dial-up, satélites, síntese de fala, linguística computacional e bioinformática. Em telecomunicação, um código convolucional é um tipo de código corretor de erro em que cada conjunto de *m* símbolos é transformado em um conjunto de *n* símbolos.

Algoritmo

❑ Inicialização:

$$\delta_1(i) = \pi_i b_i(O_1), \qquad 1 \leq i \leq N \tag{12}$$

$$\Psi_1(i) = 0$$

❑ Recursão:

$$\delta_t(j) = \max_{1 \leq i \leq N} \left[\delta_{t-1}(i) a_{ij} \right] b_j(O_t), \qquad 2 \leq t \leq T \tag{13}$$

$$\Psi_t(j) = arg \max_{1 \leq i \leq N} \left[\delta_{t-1}(i) a_{ij} \right], \qquad 1 \leq j \leq N \tag{14}$$

❑ Término:

$$P^* = \max_{1 \leq i \leq N} \left[\delta_{T(i)} \right] \tag{15}$$

$$G_T^* = arg \max_{1 \leq i \leq N} \left[\delta_{T(i)} \right] \tag{16}$$

5.1.4.3 Baum-Welch

Não existe uma maneira conhecida de resolver analiticamente o conjunto de parâmetros para um dado modelo de forma que seja maximizada a probabilidade da seqüência de observações. No entanto um procediemento iterativo como o método de Baum-Welch permite escolher $\lambda = (A, B, \pi)$ tal que $P(O|\lambda)$ é maximizada localmente. O algoritmo Baum-Welch é apresentado em termos das variáveis α_t e β_t dos algoritmos *forward* e *backward* respectivamente, e segundo Rabiner (1989) é o mais indicado para a estimação dos parâmetros do HMM. Para os HMM's discretos, a quantidade de símbolos de saída é finita. Também para uma única elocução, a re-estimação da função de probabilidade para que um estado q_i emita um símbolo $O_t = V_k$ é obtida por:

$$b_i(k) = \frac{\sum_{t=1}^{T-1} a_t(i) \beta_t(j) t.q. \quad O_t = V_k}{\sum_{t=1}^{T-1} a_t(i) \beta_t(j)}$$

$$\text{onde} \qquad b_i(k) \geq 0, \qquad 1 \leq i \leq N, \qquad 1 \leq k \leq M, \qquad \sum_{k=1}^{M} b_i(k) = 1, \qquad 1 \leq i \leq N$$

5.2 Redes Neurais Artificiais

Redes neurais artificiais são estruturas matemáticas capazes de aprender, memorizar e generalizar determinadas situações e problemas a elas apresentado. Rede neural é inspirada no cérebro e é formada por neurônios que se ligam entre si e de acordo com uma determinada função, realizam sinapses entre si (HAYKIN, 2001).

Em 1943 o psiquiatra e neuroanatomista Warren McCulloch e o matemático Walter Pitts propuseram um modelo matemático de um neurônio artificial (FAUSETT, 1994). O modelo era uma simplificação do neurônio biológico representado na Figura 5.

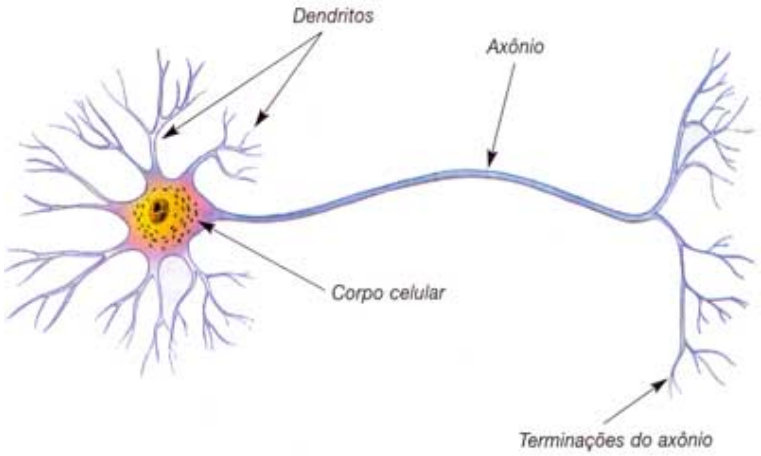

Figura 5: Estrutura básica de um neurônio biológico. *fonte:(BIOLOGIA.SEED,)*

Para representar os dendritos, o modelo usou n terminais de entrada de informações $x_1, x_2, x_3, \ldots, x_{n-1}, x_n$ e um terminal de saída y representando o axônio. As sinapses são simuladas de acordo com um coeficiente ponderador, a sinapse só ocorre quando a soma ponderada dos sinais de entrada ultrapassa um limiar pré-definido. Este limiar é chamado de função de ativação e foi definido de forma Booleana. A Figura 6 mostra o neurônio artificial.

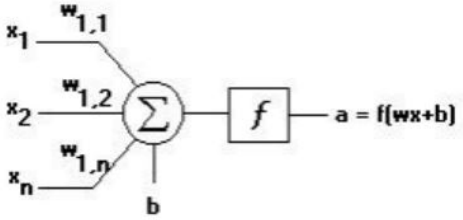

Figura 6: Neurônio artificial proposto por McCulloch e Pitts. *(FAUSETT, 1994)*

A equação 17 é a equação da saída y do neurônio.

$$y = f\Big(\sum_{i=1}^{n} x_i w_i + b\Big) \tag{17}$$

Onde n é o número de entradas do neurônio, w_i é o peso associado à entrada x_i e f é a função de ativação utilizada.

5.2.1 Classificação de Redes Neurais Artificiais (RNA)

De acordo com Haykin (2001), redes neurais podem ser classificadas de acordo com a topologia ou a forma de aprendizagem. Estas classificações são explicadas nas seções 5.2.1.1 e 5.2.1.2 respectivamente.

5.2.1.1 Topologia

As RNAs podem ser classificadas de acordo com sua topologia em perceptron de camada única, perceptron de múltiplas camadas e redes recorrentes.

1. Perceptron de Camada Única: É utilizado para classificação linear, ou seja, utilizada em problemas que sejam linearmente separáveis. A Figura 7 mostra um exemplo de perceptron de camada única;

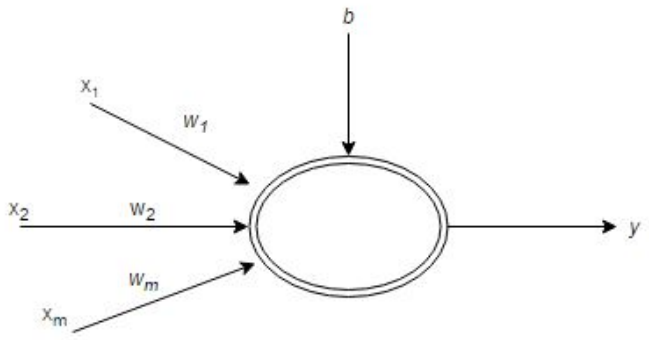

Figura 7: Perceptron de camada única.

2. Perceptron de Múltiplas Camadas: Possui várias camadas, e possuem, camadas ocultas, onde dentro delas, esta inserindo neurônios ocultos, ou seja recursos computacionais. A Figura 8 mostra um exemplo de perceptron de várias camadas;

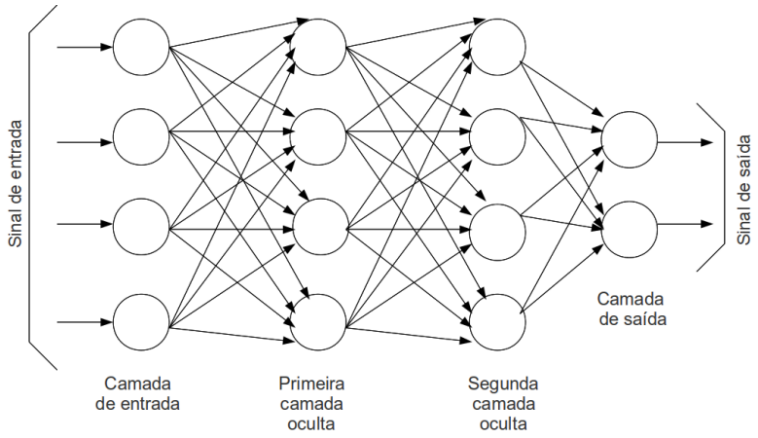

Figura 8: Perceptron de múltiplas camadas.

3. Redes Recorrentes: Possui pelo menos um laço de realimentação, pode ser implementada tanto no perceptron de camada única, como, no perceptron de multicamadas. A Figura 9 mostra um exemplo de perceptron recorrente;

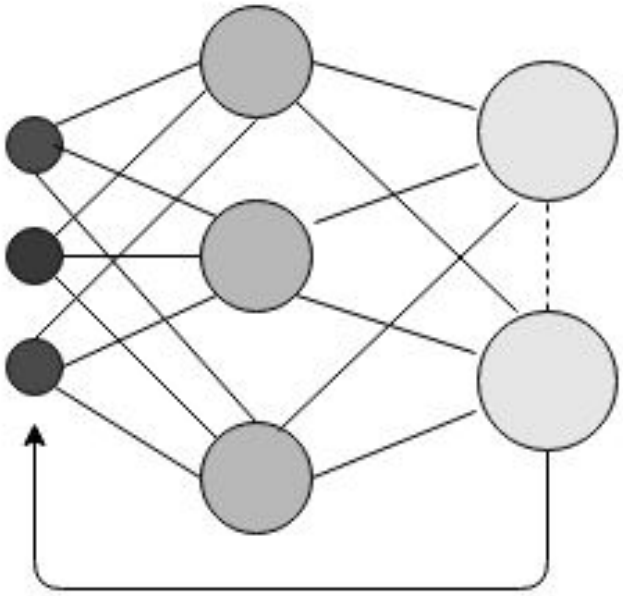

Figura 9: Perceptron recorrente.

5.2.1.2 Tipos de Aprendizagem

Outra maneira de classificar redes neurais é de acordo com o tipo de aprendizagem. Esta pode ser:

1. Aprendizagem Supervisionada: Há uma amostra que será comparada ao ambiente.

2. Aprendizagem Não Supervisionada: Procura-se um padrão, sem a ajuda de uma amostra de comparação.

5.2.2 SOM - Self Organizing Maps

De acordo com (RUSSEL, 2003) um mapa auto-orgánizavel é estruturado pelo neurônio vencedor de uma competição, ditada por uma função discriminante de cada iteração, também chamada de época. Essa competição pode ser em neurônio contra todos os neurônios da rede, ou, neurônio contra um grupo de neurônios da rede. Um das metas de um mapa auto-organizável é classificar os dados de entrada competindo entre si. O algoritmo possui um conjunto de regras de natureza local. O termo local significa que a modificação aplicada ao peso sináptico de um neurônio é confinada à vizinhança imediata daquele neurônio.

5.2.2.1 Alguns princípios intuitivos de auto-organização

A organização da rede acontece em dois níveis diferentes, que interagem entre si na forma de um laço de realimentação. De acordo com Haykin (2001) os dois niveis são:

- ❏ Atividade: Certos padrões de atividade são produzidos por uma determinada rede em resposta a sinais de entrada.
- ❏ Conectividade: Forças de conexão dos pesos sinápticos da rede são modificadas em resposta a sinais neurais dos padrões de atividade.

Pode-se citar, ainda de acordo com Haykin (2001), que mapas auto-organizáveis possuem os seguintes princípios:

- ❏ Modificações dos pesos sinápticos tendem a se auto-amplificar. Para estabilizar o sistema, deve haver alguma forma de competição por recursos limitados. Especificamente, um aumento na força de algumas sinapses da rede deve ser compensados por uma redução em outras sinapses.
- ❏ A limitação de recursos leva à competição entre sinapses e com isso à seleção das sinapses que crescem mais vigorosamente às custas das outras sinapses.
- ❏ As modificações em pesos sinápticos tendem a cooperar.
- ❏ Ordem e estrutura nos padrões de informação representam informação redundante que é adquirida pela rede neural na forma de conhecimento, que é um pré-requisito necessário para a aprendizagem.

5.2.2.2 Mapa Auto-Organizável de Kohonen

Os mapas auto-organizáveis foram desenvolvidos por Teuvo Kohonen em 1981 e fazem parte de um grupo de redes neurais baseadas em modelos de competição (FAUSETT, 1994). Uma característica importante destes mapas é que eles utilizam treinamento não supervisionado, os neurônios competem entre si e ajustam seus pesos com base nesta competição. O principal objetivo dos mapas auto-organizáveis de Kohonen é agrupar os dados de entrada que são semelhantes entre si formando classes ou agrupamentos denominados *clusters*. Segundo Affonso (2011) os mapas de Kohonen podem ser aplicados para problemas não lineares de alta dimensionalidade, como por exemplo: processamento de sinais, demodulação e transmissão de sinais, extração de características e classificação de imagens e padrões acústicos, química e medicina.

Baseado no aprendizado competitivo, os mapas de Kohonen são do tipo *winner-takes-all*, ou seja, *o vencedor leva tudo*. Os neurônios de saída competem para serem ativados e a cada iteração apenas um neurônio é ativado. O funcionamento do SOM pode ser compreendido em diferentes etapas. A etapa competitiva na qual se define o neurônio mais adequado, chamado de **BMU** (*do inglês Best Matching Unit*). A escolha da melhor correspondência entre o vetor de entrada e o vetor peso é feita por critério da menor distância euclidiana entre o vetor de pesos por ela armazenado e o vetor de entrada. Na etapa cooperativa os vizinhos são definidos dentro de uma distância obtida a partir da BMU. O processo de treinamento consiste na otimização da distância entre os neurônios. A vizinhança topológica é definida por meio da interatividade entre os neurônios. Um neurônio ativado tende a excitar os neurônios em sua vizinhança imediata. Cada iteração de atualização dos valores e distâncias da rede é chamada de **época**. As épocas constituem a etapa adaptativa.

6 IMPLEMENTAÇÃO

Neste capítulo é realizada uma discussão sobre a implementação computacional dos algoritmos estudados. O ambiente de desenvolvimento para todos os algoritmos foi o mesmo, o GNU/Linux Ubuntu 15.0.

A implementação foi dividida em etapas. A primeira etapa consiste na captura do áudio, esta foi realizada com a aplicação da biblioteca *alibsound.h*, que utiliza a interface PCM para a modulação do sinal. Nesta fase é necessário checar os parâmetros de hardware através de funções disponibilizadas pela ALSA. Após a captura da onda foi realizado o tratamento da mesma. Foram implementadas funções para aplicação da FFT, DCT, dividir o sinal em frames, aplicar janela de Hamming e por fim extrair caraterísticas do sinal. Nesta fase as principais dificuldades encontradas foram a manipulação de fórmulas matemáticas complexas e o baixo nível de programação. Após a extração das características MFCC é realizada a comparação dos padrões. A Figura 10 traz um esboço da arquitetura dos módulos desenvolvidos.

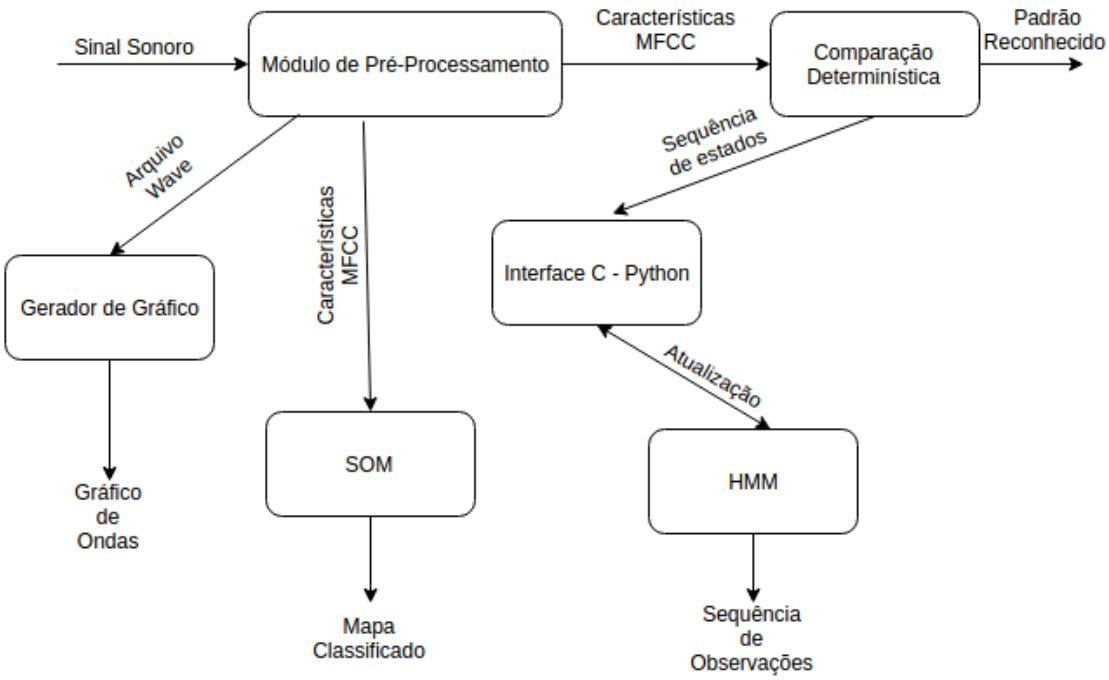

Figura 10: Organização dos módulos implementados

6.1 Bibliotecas usadas

Das bibliotecas aplicadas na implementação vamos fazer uma breve explicação daquelas que pensamos ser mais relevantes, por terem aplicações mais específicas. As bibliotecas *asoundlib.h* e *dirent.h* da linguagem de programação C foram usadas, a primeira

para a captura do áudio, como já foi citado no capítulo 3, e a segunda para a manipulação das palavras e arquivos armazenados. A biblioteca *asoundlib.h* é disponibilizada pela API ALSA. A ALSA pode ser dividida de acordo com as interfaces suportadas por ela. Dentre estas interfaces foi usada a interface para o gerenciamento de captura de áudio digital e reprodução, a interface PCM. Programas que usam esta interface geralmente usam a estrutura mostrada no algoritmo 1.

Algoritmo 1 Formato de programa ALSA

```
1: Abra Interface para captura ou reprodução
2: Setar parâmetros de hardware
3: enquanto existirem dados a serem processados faça
4:     Leia dado PCM
5:     Escreva dado PCM
6: fim enquanto
7: Fecha Interface
```

Em (KYSELA et al., 2014) é possível encontrar uma explicação masi detalhada sobre a biblioteca *alibsound.h* e a manipulação dos parâmetros de hardware.

6.2 Comparação de Padrões

Após a extração das caraterísticas MFCC pode-se realizar a comparação entre o padrão de entrada e os padrões armazenados. Esta comparação pode ser realizada por diferentes algoritmos. Neste trabalho usamos um método determinístico, um método de inteligência artificial (SOM) citado em 5.2.2 e um método estocástico (HMM), apresentado na seção 5.1.

6.2.1 Método Determinístico

O método de comparação de padrão determinístico compara a correlação entre dois vetores de características MFCC, quanto menor o resultado retornado pela função mais parecidos são os padrões. Este algoritmo foi implementado em linguagem C. Neste método é usado um limiar de correlação para definir se os padrões são iguais, ou suficientemente parecidos. O algoritmo 3 é usado para o cálculo da correlação entre os vetores MFCC.

6.2.2 Método baseado em inteligência artificial - SOM

O método baseado em inteligência artificial implementado foi o mapa auto-organizável, ou SOM. A implementação do SOM foi realizada em linguagem de

Algoritmo 2 Correlação entre vetores MFCC

```
1: função COMPARE(vetor1, vetor2)
2:     para i ← 1 até n1 faça                          ▷ n1 é a quantidade de frames do vetor 1
3:         para j ← 1 até n2 faça                      ▷ n2 é a quantidade de frames do vetor 2
4:             para k ← 1 até N_MFCC faça              ▷ N_MFCC é a quantidade de coeficientes
5:                 dist[i][j] ← (vetor1[i].feature[k] − vetor2[i].feature[k])^2
6:             fim para
7:             dist[i][j] ← √dist[i][j]
8:         fim para
9:     fim para
10:    para i ← 1 até n1 faça
11:        para j ← 1 até n2 faça
12:            se dist[i][j] < min então
13:                min ← dist[i][j]
14:            fim se
15:            dist[i][j] ← dist[i][j] + min
16:        fim para
17:    fim para
18:    devolve dist[n1][n2] / √(n1^2+n2^2)
19: fim função
```

programação C. A principal dificuldade encontrada na implementação deste método foi a manipulação do mapa e seus neurônios. O gerenciamento da memória na alocação e liberação de vários ponteiros também se mostrou bastante complexa. O mapa auto-organizável requer mais memória e tempo na execução do que o método determinístico citado na 6.2.1.

Algoritmo 3 SOM

```
1: função COMPARE(vetor1, vetor2)
2:     Inicialize os pesos com valores aleatórios
3:     enquanto épocas < 1000 faça
4:         Apresente o padrão de entrada à rede
5:         Escolha o neurônio de saída com maior estado de ativação
6:         Atualize os pesos dos neurônios vizinhos ao neurônio vencedor
7:         Reduza o fator de aprendizado linearmente
8:     fim enquanto
9: fim função
```

O mapa foi montado com entradas apresentadas a ele, uma amostra de cada padrão a ser reconhecido, após a inicialização do mapa é realizado o treinamento apresentando várias amostras diferentes de cada padrão. A saída do algoritmo é um mapa com a classificação das entradas.

6.2.3 Método estocástico - HMM

O algoritmo HMM foi implementado em linguagem computacional Python 2.7. Python é uma linguagem de tipagem dinâmica de alto nível, isto facilta na manipulação das estruturas de dados usadas pelo algoritmo. O algoritmo 4 mostra os passos para resolver o problema da melhor sequência de estados, como mostrado na seção 5.1.4.2.

Algoritmo 4 Viterbi

```
1:  função VITERBI(hmm, distribuição inicial, emissões)
2:      Inicialize a probabilidade das emissãoes iniciais
3:      para i ← em até emissões faça
4:          probabilidadedetransio ← probabilidadedetransio ∗ prob.emissocorrente)
5:          max ← argmax(probabilidadedetranmsio)
6:          probs ← dist.deemisses ∗ probabilidadedetransiomxima
7:          insira o maior resultado na pilha
8:      fim para
9:      sequnciadeestados ← oestadoquegerouamaiorprobabilidadedeobservao
10:     enquanto houver estados na pilha faça
11:         retire estados da pilha
12:         sequnciadeestados ← estadocommaiorprobabilidadedeobservao
13:     fim enquanto
14:     inverta sequência de estados
15:     return sequência de estados
16: fim função
```

7 RESULTADOS

Foram estudados e avaliados três diferentes abordagens para o reconhecimento de palavras, sempre levando em consideração o contexto da aplicação e o caso de teste. O trabalho foi dividido em duas etapas, na primeira etapa ocupou-se da captura e tratamento do sinal e extração de características, já na segunda etapa foram avaliados os métodos de comparação de padrões. Foram avaliados um método estocástico (HMM), um método determinístico (correlação entre vetores) e um método de inteligência articial (SOM).

Os testes foram realizados para o seguinte vocabulário: ***ajuda, assalto, ladrão, polícia, socorro***. Estas palavras foram escolhidas com base na proposta de testar os algoritmos para monitoramento de ambientes. Podemos ressaltar que, para a proposta, a precisão de acertos não precisava ser alta, ou seja, falsos positivos são toleráveis. Foram gravadas 50 amostras de cada palavra como padrões de comparação. As entradas para reconhecimento foram apresentadas ao sistema dez vezes em um ambiente livre de ruídos e dez vezes em um ambiente ruidoso.

Tabela 3: Taxa de acertos do método determinístico isolado

Palavra	Ambiente Silencioso	Ambiente Ruidoso
Ajuda	70%	60%
Assalto	80%	40%
Ladrão	60%	34%
Polícia	90%	60%
Socorro	90%	80%

A tabela 3 mostra a porcentagem de acerto para cada palavra usando o método determinístico. Cada palavra foi inserida dez vezes por um número aleatório de pessoas em um ambiente com pouco ruído a média da taxa de acertos foi de 78% e em um ambiente com muito ruído esta taxa caiu para 66,4%.

Tabela 4: Taxa de acertos do método determinístico contínuo

Palavra	Isolado	Contínuo
Ajuda	60%	35%
Assalto	40%	35%
Ladrão	30%	34%
Polícia	60%	37%
Socorro	80%	40%

A tabela 4 mostra a porcentagem de acerto para cada palavra usando o método determinístico com abordagem contínua. Cada palavra foi inserida vinte vezes por um número aleatório de pessoas em um ambiente ruidoso. A média da taxa de acertos foi de 45,1% .

Tabela 5: Taxa de acertos do método SOM

Palavra	Ambiente Silencioso	Ambiente Ruidoso
Ajuda	70%	65 %
Assalto	80%	55 %
Ladrão	75%	65 %
Polícia	75 %	65%
Socorro	80%	70%

A tabela 5 mostra a porcentagem de acerto para cada palavra usando o algoritmo SOM. Cada palavra foi inserida dez vezes por um número aleatório de pessoas em um ambiente com pouco ruído a média da taxa de acertos foi de 76% e em um ambiente com muito ruído esta taxa caiu para 64%.

Tabela 6: Comparação entre os métodos

–	Determinístico	HMM	SOM
Taxa de acerto	66,4%	–	70%
Tempo de execução	rápido	viciada	médio
Custo computacional	baixo	alto	alto

A tabela 6 traz uma comparação entre os métodos estudados. O algorimto HMM, como foi explicado no capítulo 5.1, é baseado em probabilidade e nas transições de um estado a outro. Para garantir a homogeneidade do sistema os valores de transição dos estados são iguais

isso gera um vício. Sempre que uma palavra é identificada não ocorre transição de estados, retornando sempre o estado anterior.

Para o caso estudado neste trabalho podemos concluir que o método determinístico foi o que se mostrou mais eficiente, pois este consome menos recursos computacionais e possui uma taxa de acerto satisfatória.

8 CONCLUSÃO

O presente trabalho teve como objetivo estudar algoritmos para reconhecimento de palavras isoladas em fluxo contínuo. Os métodos estudados deveriam ter a capacidade de reconhecer uma palavra isolada independente de locutor, livre de contexto em um ambiente qualquer. Também deveria ser considerado os rescursos computacionais usados. Uma solução que requer pouca memória e com rápido tempo de processamento. Com base nisto podemos concluir que o método determinístico foi o que apresentou melhores resultados, pois o tempo de execução deste é baixo e requer poucos recursos computacionais podendo ser utilizado em dispositivos com baixo poder de processamento.

Durante a etapa de captação e processamento do sinal a qualidade do hardware usado tem grande impacto sobre as caratecterísticas extraídas. Em um sistema ideal os filtros devem ser implementados em hardware o que garante maior processamento e robustez aos ruidos. As caraterísticas usadas também devem ser analisadas, métodos como ***PNCC, RASTA-PLP, PLP, LPC, DBNF*** são exemplos de descritores de sinal, tal como MFCC, e podem ser analisados em trabalhos futuros.

A comparação entre o padrão buscado e os padrões armazenados é uma fase independente, levando em consideração as características usadas para o pré-processamento do sinal, ou seja, é possível aplicar diferentes algoritmos nesta fase. As técnicas aplicadas durante todo o processo de reconhecimento que classificam o sistema de reconhecimento. Geralmente são empregados redes neurais artificiais ou algoritmos probabilísticos, como o HMM. Também são aplicadas técnicas determinísticas onde a classificação é realizada através de fórmulas matemáticas, estas possuem processamento rápido, porém são mais dependentes da variação do sinal (timbre de voz, altura da voz) e sensíveis ao ruído.

Por fim gostariamos de sugerir para trabalhos futuros, além dos descritores de voz citados acima, a implementação de um algoritmo para detecção de atividade de voz como citado em (BORGES, 2008), RNAs multicamadas e o Modelo de Misturas Gaussianas (GMM).

REFERÊNCIAS BIBLIOGRÁFICAS

AFFONSO, G. S. **Mapas Auto- organizáveis de Kohonen (SOM) aplicados na avaliação dos parâmetros da qualidade da água.** Dissertação (Mestrado) — USPE, 2011.

BIOLOGIA.SEED. **Mecanismos biológicos**. Disponível em: <http://www.biologia.seed.pr.gov.br/modulos/galeria>. Acesso em: 29.09.2016.

BORGES, R. A. **Implementação e Comparação de algoritmos de detecção de atividade de voz (VAD) em DSP**. [S.l.: s.n.], 2008.

FAUSETT, L. V. **Fundamentals of Neural Networks: Architectures Algorithms and Applications.** [S.l.]: Prentice-Hall, 1994.

GORDILLO, C. D. A. **Reconhecimento de Voz Contínua Combinando os Atributos MFCC e PNCC com Métodos de Robustez SS, WD, MAP e FRN**. Dissertação (Mestrado) — PUC-RJ, 2013.

HAYKIN, S. **Redes Neurais, Princípios e Prática.** [S.l.]: Ed. Bookman, 2001.

INCER, A. N. **Digital Speech Processing, Speech Coding, Syntesis and Recognition**. [S.l.]: Kluwer Academic Publishers, 1992.

KYSELA, J. et al. **ALSA project - the C library reference**. 2014. Acessado em: 28 Setembro, 2016. Disponível em: <http://www.alsa-project.org/alsa-doc/alsa-lib/>.

MORGAN, D. P.; SCOFIELD, C. L. **Neural Networking and Speech Processing**. [S.l.]: Kluwer Academic Publishers, 1991.

ORTIGUEIRA, M. D. **Processamento Digital de Sinais**. [S.l.]: Fundação Calouste Gulbenkian: Lisboa, 2005.

PROAKIS, J. G.; MANOLAKIS, D. G. **Digital Signal Processing. Principles, Algorithms and Applications**. [S.l.]: Prentice-Hall: New Jersey, 1996.

RABINER, L. R. A tutorial on hidden markov models and selected aplications in speech recognition. **Proceedings of the IEEE.**, p. Vol. 77 – 2, 1989.

RABINER, L. R.; JUANG, B. H. **Fundamentals of Speech Recongnition**. [S.l.]: Prentice-Hall, 1993.

RUSSEL, S. J. **Artificial Intelligence: A Modern Approach.** [S.l.]: Pearson, 2003.

TRANTER, J. Introduction to sound programming with alsa. **Linux Journal**, 2004. Disponível em: <http://www.linuxjournal.com/article/6735>. Acesso em: 10.4.2015.

VIANA, H. O. **Descritor de voz invariante ao ruído.** Dissertação (Mestrado) — UFPE, Centro de Informática, Programa de Pós-graduação em Ciência da Computação, 2013.