UNIVERSIDADE DE TAUBATÉ

Rodrigo Jorge Alvarenga

Reconhecimento de Comandos de Voz por Redes Neurais

Taubaté – SP

2012

Rodrigo Jorge Alvarenga

Reconhecimento de Comandos de Voz por Redes Neurais

Tese apresentada para obtenção do Certificado de Título de Mestre pelo Curso Mestrado em Engenharia Mecânica do Departamento de Engenharia Mecânica da Universidade de Taubaté, Área de Concentração: Automação e Controle
Orientador: Prof. Dr. Pedro Paulo Leite do Prado

Taubaté – SP

2012

RODRIGO JORGE ALVARENGA

RECONHECIMENTO DE COMANDOS DE VOZ POR REDES NEURAIS

Tese apresentada para obtenção do Certificado de Título de Mestre pelo Curso Mestrado em Engenharia Mecânica do Departamento de Engenharia Mecânica da Universidade de Taubaté, Área de Concentração: Automação e Controle
Orientador: Prof. Dr. Pedro Paulo Leite do Prado

Data: _____

Resultado: _____

BANCA EXAMINADORA

Prof .Dr. Pedro Paulo Leite do Prado - Universidade de Taubaté

Assinatura_____

Prof. Dr. Sebastião Cardoso - Vale Soluções em Energia (VSE)

Assinatura_____

Prof. Dr. Álvaro Manoel de Souza Soares – Universidade de Taubaté

Assinatura_____

Dedico esta tese aos meus pais, Romualdo e Célia, que sempre priorizaram minha formação, me apoiaram e me incentivaram a crescer cada vez mais buscando realizar os meus sonhos.

AGRADECIMENTOS

Ao Professor Dr. Pedro Paulo Leite do Prado, por sua excelente orientação, paciência e amizade;

Aos integrantes da banca, pelas sugestões que ajudaram na melhoria do trabalho;

À minha tia Dra. Lília Maíse de Jorge, pelo seu grande apoio e por sua solícita ajuda;

A todos os amigos e familiares que me apoiaram e colaboraram com o trabalho, fornecendo amostras para análise.

RESUMO

Sistema de reconhecimento de fala tem amplo emprego no universo industrial, no aperfeiçoamento de operações e procedimentos humanos e no setor do entretenimento e recreação. O objetivo específico do trabalho foi conceber e desenvolver um sistema de reconhecimento de voz, capaz de identificar comandos de voz, independentemente do locutor. A finalidade precípua do sistema é controlar movimentos de robôs, com aplicações na indústria e no auxílio de deficientes físicos. Utilizou-se a abordagem da tomada de decisão por meio de uma rede neural treinada com as características distintivas do sinal de fala de 16 locutores. As amostras dos comandos foram coletadas segundo o critério de conveniência (em idade e sexo), a fim de garantir uma maior discriminação entre as características de voz, e assim alcançar a generalização da rede neural utilizada. O pré-processamento consistiu na determinação dos pontos extremos da locução do comando e na filtragem adaptativa de Wiener. Cada comando de fala foi segmentado em 200 janelas, com superposição de 25% . As *features* utilizadas foram a taxa de cruzamento de zeros, a energia de curto prazo e os coeficientes ceptrais na escala de frequência *mel*. Os dois primeiros coeficientes da codificação linear preditiva e o seu erro também foram testados. A rede neural empregada como classificador foi um *perceptron* multicamadas, treinado pelo algoritmo *backpropagation*. Várias experimentações foram realizadas para a escolha de limiares, valores práticos, *features* e configurações da rede neural. Os resultados foram considerados muito bons, alcançando uma taxa de acertos de 89,16%, sob as condições de pior caso da amostragem dos comandos.

Palavras-chave:

Automação. Processamento de Sinais. Reconhecimento de palavras. MFCC. Coeficientes "*mel-cepstral*". LPC. Redes Neurais. *Backpropagation*.

ABSTRACT

Systems for speech recognition have widespread use in the industrial universe, in the improvement of human operations and procedures and in the area of entertainment and recreation. The specific objective of this study was to design and develop a voice recognition system, capable of identifying voice commands, regardless of the speaker. The main purpose of the system is to control movement of robots, with applications in industry and in aid of disabled people. We used the approach of decision making, by means of a neural network trained with the distinctive features of the speech of 16 speakers. The samples of the voice commands were collected under the criterion of convenience (age and sex), to ensure a greater discrimination between the voice characteristics and to reach the generalization of the neural network. Preprocessing consisted in the determination of the endpoints of each command signal and in the adaptive Wiener filtering. Each speech command was segmented into 200 windows with overlapping of 25%. The features used were the zero crossing rate, the short-term energy and the mel-frequency ceptral coefficients. The first two coefficients of the linear predictive coding and its error were also tested. The neural network classifier was a multilayer perceptron, trained by the backpropagation algorithm. Several experiments were performed for the choice of thresholds, practical values, features and neural network configurations. Results were considered very good, reaching an acceptance rate of 89,16%, under the "worst case" conditions for the sampling of the commands.

Keywords: Automation. Signal Processing. Word Recognition. MFCC. Mel-frequency Ceptral Coefficients. LPC. Neural Networks. Backpropagation.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS E ABREVIATURAS

ADALINE - Adaptive Linear Element

ANN - Artificial Neural Network

AT&T - American Telephone and Telegraph

AV-ASR - Audio -Visual Automatic Speech Recognition

BAM - Bidirectional Associative Memory

CCS C - Custom Computer Services (Compilador C)

DCT - Discrete Cosine Transform

FIR - Finite Impulse Response

HMM - Hidden Markov Model

IBM - International Business Machines

LED - Light Emitter Diode

LSM - Least Square Method

LVQ - Learning Vector Quantization

MADALINE: Multiple ADALINE

MLP - Multi Layer Perceptron

MFCC - Mel-frequency Cepstral Coefficents

OCR - Optical Character Recognizer

PCM - Pulse Code Modulation

PIC - Peripheral Interface Controller

RNA - Redes Neurais Artificiais

SOM - Self-Organization Maps

SVD - Singular Value Decomposition

TTS - Text-to-Speech

VQ - Vector Quantization

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO

1.1. Histórico e Estado da Arte

Sistemas de reconhecimento de fala por máquina têm sido o foco de estudo, já há algumas décadas, de profissionais de diversas áreas, sobretudo os de engenharia e computação. Trata-se de um recurso de amplo espectro de uso no:

- universo industrial, por exemplo, no controle de máquinas e robôs;

- campo do aperfeiçoamento e melhoria de operações e procedimentos humanos, por exemplo, na conversão automática de fala contínua para texto;

- setor do entretenimento e recreação, exemplificado pelo comando de brinquedos, e em outras áreas.

De acordo com Rabiner e Juang (1993), pesquisas em processamento de fala são necessárias, porque ainda falta muito para se atingir um nível ideal de interação máquina/homem, no qual a máquina possa compreender o que é dito por qualquer locutor em todos os ambientes possíveis.

Os sistemas de reconhecimento de fala podem ser classificados quanto a três parâmetros diferentes:

- dependência de locutor: o reconhecimento pode ser independente ou dependente do locutor;

- tipo de locução: palavras isoladas ou locuções contínuas;

- tamanho do vocabulário.

O início dos estudos sobre este assunto ocorreu na década de 1950, com pesquisas sobre reconhecimento de fonemas acústicos, reconhecimento de dígitos proferidos por um único locutor e reconhecimento de sílabas. Os pesquisadores da época utilizavam *software,* que apesar de inovadores, apresentavam problemas nas medições de ressonâncias vocálicas. No final dessa década já há relatos de estudos que incorporavam informações estatísticas na ordenação de fonemas permitidos.

Na década de 1960 houve investimentos em projetos de *hardware* para reconhecimento de voz, especialmente no Japão. Em 1962, a Universidade de Kyoto projetou um *hardware* capaz de reconhecer fonemas utilizando um analisador de cruzamento de zeros. Ao final dos anos 60 ocorreram as pioneiras pesquisas de reconhecimento de fala contínua (RABINER; SCHAFER, 1978).

A década seguinte foi marcada pelo desenvolvimento de sistemas para reconhecimento de palavras isoladas e estudos sobre o uso da teoria de reconhecimento de padrões para aplicações em reconhecimento de fala. Iniciaram-se também os estudos na IBM (*International Business Machines*) e na empresa americana AT&T (*American Telephone and Telegraph*) (RABINER; JUANG, 1993).

Nos anos 80, as pesquisas passaram a contar com a utilização da tecnologia de redes neurais (RABINER; JUANG, 1993).

Nos últimos anos, o enfoque tem sido no o reconhecimento de fala contínua utilizando sistemas híbridos, que empregam dois ou mais processos, em diferentes fases do reconhecimento, visando a uma melhoria de desempenho. O estado da arte consiste nesse tipo de combinação das técnicas de HMM (*Hidden Markov Model*), MFCC (*Mel-frequency Cepstral Coefficents*) e ANN (*Artificial Neural Networks*) (RABINER; JUANG, 1993).

Uma nova direção da pesquisa é a integração do reconhecimento da fala ao reconhecimento visual, a fim de atingir taxas de erro cada vez menores. Esses sistemas são conhecidos como AV-ASR (*Audio-Visual Automatic Speech Recognition*) e utilizam leitura labial como complemento do reconhecimento de fala.

Existem atualmente estudos realizados em bancos de locuções, de larga escala, que objetivam comparar a taxa de erro de reconhecimento de fala alcançada por ouvintes humanos e por máquinas.

O reconhecimento de fala por máquina é um desafio computacional, pois o sinal digital de fala é dinâmico e sempre apresenta uma estrutura diferente, ainda que seja pronunciado pelo mesmo locutor.

O cérebro humano pode cumprir esta tarefa com certa facilidade devido à grande base de experiências adquiridas pelo ser humano desde o seu nascimento. Analogamente, em sistemas digitais, o uso de redes neurais artificiais, com base em adequadas características distintivas da fala, é capaz de criar uma base de conhecimentos, menor que a do cérebro humano, porém suficiente para que sistemas de reconhecimento de padrão atinjam seus objetivos com sucesso.

1.2. Objetivos

O objetivo geral desse trabalho é contribuir para o desenvolvimento de sistemas de reconhecimento de fala, visando ao aperfeiçoamento da interação homem/máquina.

O seu objetivo específico é desenvolver um sistema de reconhecimento de voz, capaz de identificar comandos de voz, independentemente do locutor. A finalidade precípua do

sistema é controlar movimentos de robôs, com aplicações na indústria (comando de máquinas) e no auxílio de deficientes físicos.

As contribuições adicionais consistem no fornecimento de subsídios a respeito de:

- processamento de sinais;
- características principais do sinal de fala;
- reconhecimento de padrões;
- redes neurais.

1.3. Metodologia

A metodologia utilizada para alcançar os objetivos propostos apresenta um delineamento teórico-experimental. A abordagem utilizada foi a da tomada de decisão por meio de uma rede neural treinada com as características distintivas do sinal de fala de diversos locutores.

A base teórica é a das técnicas de processamento de sinais, reconhecimento de padrões e redes neurais.

A experimentação é orientada no sentido de obter a menor taxa de erros possível no reconhecimento dos comandos. Consiste na busca de:

- valores ótimos dos limiares de decisão entre sinal e ruído na fase de pré-processamento;
- características distintivas (*features*) do sinal de fala de maior poder de discriminação;
- arquiteturas de redes neurais mais eficientes.

As amostras dos comandos foram coletadas segundo o critério de conveniência (em idade e sexo), com o objetivo de garantir uma maior discrepância entre as características de voz dos participantes e assim alcançar a generalização da rede neural utilizada no sistema.

Dois grupos distintos participaram da coleta de dados: o de treinamento e o de teste.

1.4. Revisão da Literatura

As pesquisas em redes neurais artificiais começaram na década de 1940 (BRAGA; CARVALHO; LUDEMIR, 2007).

Em 1943, McCulloch e Pitts (1943 apud HAYKIN, 1999) fizeram uma analogia entre neurônios biológicos e o processo eletrônico, estabelecendo as bases da neurocomputação.

Em 1949, Hebb (1949 apud BRAGA; CARVALHO; LUDEMIR, 2007) sugeriu que neurônios frequentemente ativados em conjunto são ligados em organizações funcionais chamadas "estruturas celulares" e "sequências em fase", as quais, quando estimuladas, correspondem à invocação de uma ideia elementar. As sinapses deveriam ter seu valor (peso) atribuído dinamicamente, de acordo com o uso, ou seja, as redes neurais artificiais poderiam aprender (BRAGA; CARVALHO; LUDEMIR, 2007).

No final da década de 1950, Rosemblatt (1950 apud BRAGA; CARVALHO; LUDEMIR, 2007), na Universidade de Cornell, em continuação aos estudos de McCulloch, criou uma rede de múltiplos neurônios que chamou de "*Perceptron*", como modelo para reconhecimento de padrões visuais.

Na mesma época em que Rosemblatt trabalhava no "*Perceptron*", Widrow (1960 apud BRAGA; CARVALHO; LUDEMIR, 2007), na Universidade de Stanford, desenvolveu um modelo neurolinear chamado "ADALINE" (*Adaptive Linear Element*), em que tentava simular o cérebro humano com processadores paralelos. Mais tarde realizou a sua generalização multidimensional, o MADALINE (*Multiple* ADALINE).

Em 1969, Minsky e Papert (1969 apud BRAGA; CARVALHO; LUDEMIR, 2007) publicaram críticas ao *Perceptron* e ao ADALINE que causaram um grande choque no meio científico. As críticas baseavam-se na impossibilidade de uma rede de um único nível, como o *Perceptron* e o ADALINE*,* aprender um simples padrão, como o da função lógica ou-exclusivo, por exemplo.

Durante a década de 1970 James Anderson (1977 apud BRAGA; CARVALHO; LUDEMIR, 2007) desenvolveu a rede neural artificial *Brain-State-in-a-Box*. Mas de certa forma, nesta época, houve pouca divulgação de pesquisas em redes neurais.

Já na década de 1980, essas pesquisas ressurgiram com interesse renovado devido ao desenvolvimento do algoritmo *backpropagation* (retropropagação), que permitiu o treinamento de redes multicamadas, resolvendo o problema apresentado por Minsky e Papert (1969 apud CARRARA, 1997), e devido à popularização do trabalho de Hopfield (1982 apud CARRARA, 1997) para resolver problemas de otimização, utilizando redes realimentadas (CARRARA, 1997).

Ao mesmo tempo Kosko (1987 apud HAYKIN 1999) desenvolveu várias linhas de pesquisa que abordavam memórias associativas bidirecionais (BAM, *Bidirectional Associative Memory*), mapas cognitivos nebulosos e a memória associativa nebulosa (HAYKIN, 1999; MATLAB, 2000).

Um ano depois, Hinton e Fahlman apresentaram a arquitetura da máquina de Boltzmann, uma rede de Hopfield que utiliza o método de redução do erro *Simulated Annealing,* para escapar de mínimos locais (BRAGA; CARVALHO; LUDEMIR, 2007).

Para o emprego de redes neurais, neste trabalho, foram consultados o livro de Haykin (1999) e Katagiri (2000) para a apresentação dos fundamentos; o *Neural Network Toolbox* do MATLAB®, (BEALE; HAGAN; DEMUTH, 2011) e o livro de Braga, Carvalho e Ludemir (2007) para a implementação do algorítmo *Backpropagation*.

Os fundamentos de processamento de sinais e reconhecimento de padrões foram adquiridos nos livros de Oppenhein e Schafer (1989) e Tou e Gonzalez (1981).

As principais obras consultadas que tratam do processamento de sinais de fala foram Rabiner e Schafer (1978) e Rabiner e Juang (1993).

A principal referência para a aplicação dos coeficientes MFCC (*Mel-frequency Cepstral Coefficients*) foi a (SKOWRONSKI; HARRIS, 2002).

De muita utilidade foi a dissertação de Bezerra (1994), para o pré-processamento do sinal de fala e extração das suas características principais.

1.5. Estrutura do Trabalho

O Capítulo 1 resume o histórico e estado da arte em reconhecimento de padrões de fala, os objetivos do trabalho, a metodologia geral e as principais referências bibliográficas pesquisadas.

O Capítulo 2 aborda o conceito de redes neurais, sua classificação quanto à estrutura e tipos de aprendizagem, as principais arquiteturas utilizadas e o algoritmo *backpropagation.*

O Capítulo 3 apresenta os fundamentos de reconhecimento de sinal de fala, as fases do reconhecimento de padrões, a produção da fala e as características distintivas principais (*features*) empregadas no desenvolvimento do reconhecedor.

O Capítulo 4 descreve o reconhecedor de comandos de voz à base de redes neurais, apresentando os procedimentos para a coleta de dados (locutores, instrumentos, condições etc.), pré-processamento, extração das características principais (*features*) e classificação, os resultados obtidos e a contribuição do trabalho.

O Capítulo 5 realiza a discussão dos resultados e apresenta as conclusões finais. Para finalização, seguem as referências.

CAPÍTULO 2 - FUNDAMENTOS DE REDES NEURAIS ARTIFICIAIS

2.1. Introdução

Neste capítulo são apresentados os conceitos fundamentais a respeito de redes neurais, com ênfase nos pontos empregados na técnica de reconhecimento de padrões de fala utilizada nesta dissertação.

Redes neurais artificiais (RNA) são sistemas que tentam simular a estrutura do cérebro humano para solucionar problemas ou realizar tarefas que seriam complexas para outros sistemas computacionais. Esses sistemas são considerados inteligentes, pois empregam algoritmos de aprendizagem que permitem a adaptação do sistema no intuito de produzir a saída desejada.

Uma rede neural é formada por unidades chamadas de nodos ou neurônios que são inspirados na estrutura do neurônio biológico. Os neurônios trabalham paralelamente e são conectados entre si. Cada nodo realiza um processamento e passa o resultado ao nodo seguinte.

2.2. Neurônio biológico

O neurônio biológico é composto basicamente por três partes: dendritos, axônios e corpo da célula, conforme mostra a Fig.2.1 Os dendritos são responsáveis por conduzir informações (impulsos nervosos) oriundas de outros neurônios ou do meio externo. No corpo da célula é realizado o processamento dos impulsos gerando novos impulsos. Estes novos impulsos são transmitidos aos neurônios seguintes através dos axônios. A conexão entre um dendrito e um axônio, ilustrada na Fig.2.2, é chamada de sinapse.

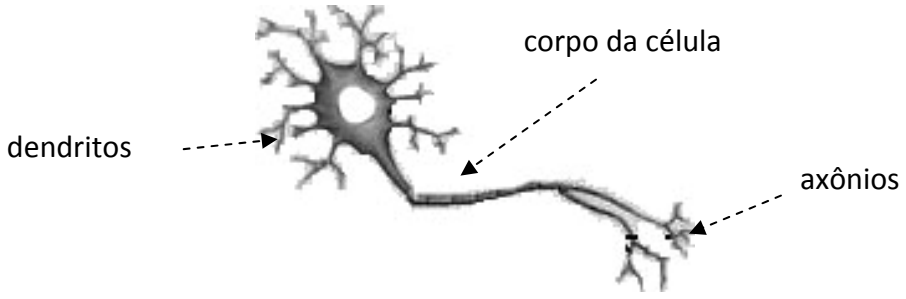


Fig. 2.1 Componentes de um Neurônio

As sinapses trabalham como válvulas que controlam o fluxo da informação (KATAGIRI, 2000). Ela é processada no corpo do neurônio e caso atinja um limiar requerido, um novo impulso será gerado.

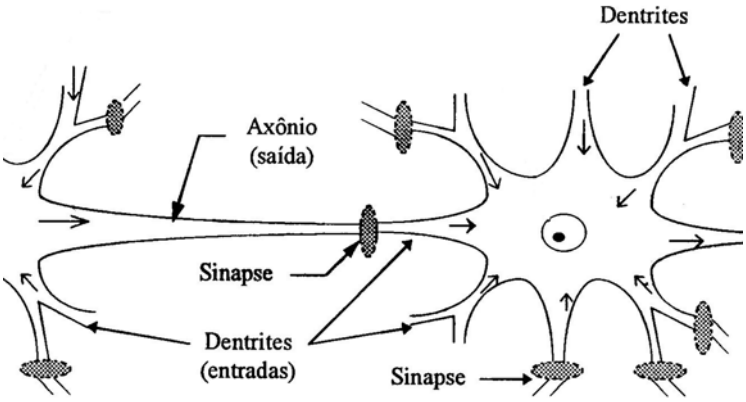


Fig. 2.2 Sinapses entre Neurônios

Um modelo simplificado do neurônio biológico mostrado na Fig.2.3 foi proposto por McCulloch e Pitts em 1943 (BEALE et al., 2011). Esses autores consideravam que todo fenômeno psicológico era passível de ser analisado e compreendido por meio de uma rede de dispositivos lógicos de apenas dois estados que variam de acordo com suas atividades de entrada.

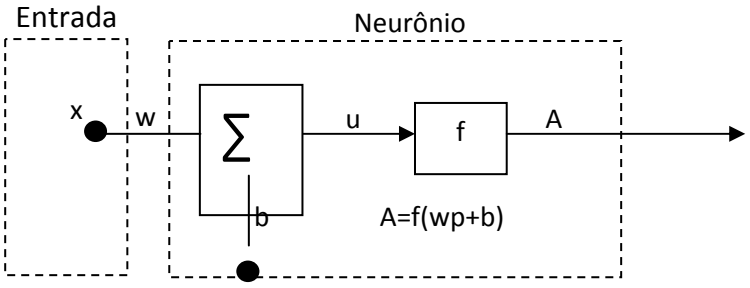


Fig.2.3 Modelo do Neurônio de McCulloch e Pitts

Cada dendrito possui um peso que representa o quanto o valor da entrada será significativo para o neurônio. Assim, cada entrada **x** é multiplicada por um peso **w** e somada a uma polarização (*bias*) **b**, resultando no valor **u = wx+b**, que então é submetido a uma função de transferência **f,** cuja saída é **A.** Se **y** for superior a um limiar pré-determinado, então o neurônio terá uma saída ativa.

2.3. Rede Neural Elementar

Com base no modelo de McCulloch e Pitts, considerando **n** entradas e **m** saídas, é organizada como exemplo uma rede neural elementar, com uma camada de entrada, uma camada de saída, sem polarizações e uma função de transferência linear, conforme Fig.2.4.

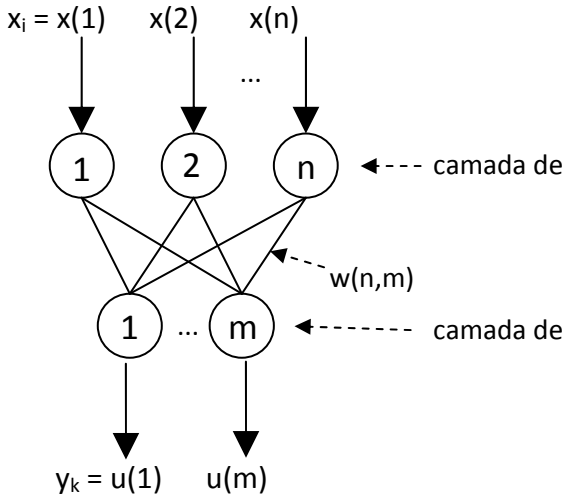


Fig. 2.4. Rede Neural Elementar

A equação dessa rede neural elementar pode ser expressa por:

$$[y_k]_{1,m} = [x_i]_{1,n} \; [W_{i,k}]_{n,m} \tag{2.1}$$

onde:

$[y_k]_{1,m}$: vetor de saída , k = 1,2..m;

$[x_i]_{1,n}$ vetor de entrada, i = 1,2..n;

$[W_{i,k}]_{n,m}$: matriz de pesos;

A função de erro ou função objetiva, pelo método dos mínimos quadrados (LSM – *Least Square Method*, ou Regra Delta), é expressa por:

$$E_T = \sum_{k}^{m} (y_k - T_k)^2 \tag{2.2}$$

onde:

E_T = erro total pelo método dos mínimos quadrados;

$[y_k]_{1,m}$: vetor de saída, k = 1,2..m;

T_K: alvos ou valores desejados, k = 1,2..m.

2.4. Treinamento

O treinamento pode ser supervisionado ou não supervisionado (HAYKIN, 1999).

No treinamento supervisionado os pesos e polarizações são alterados de conformidade com um determinado algoritmo até que o erro total atinja o limiar estabelecido. O limiar deve ser estabelecido experimental e judiciosamente para cada classe de problema, a fim de evitar o chamado supertreinamento, quando se diz que a rede "memorizou" os dados, perdendo a capacidade de generalização.

Há dois modos de operacionalizar as variações cíclicas dos pesos e polarizações:

- Modo sequencial ou padrão: as entradas do conjunto de treinamento e os respectivos alvos são apresentados sucessivamente à rede. Para cada par "vetor de entrada / alvo" é feita a atualização dos pesos e polarizações. Quando todas as entradas tiverem sido apresentadas estará configurado um ciclo (ou *epoch*). As *epochs* continuam até que seja alcançada a especificação para o erro.

- Modo *batch* (batelada): todas as entradas do conjunto de treinamento e respectivos alvos são apresentados simultaneamente à rede.

No treinamento não supervisionado não são fornecidas as entradas e as respectivas saídas desejadas. É realizado o "mapeamento de auto-organização". As amostras dos fenômenos (pontos no espaço de atributos do evento) são agrupadas por algoritmos de "*clustering*" (*Maximim-distance*, K-means, Isodata etc.) e são aplicadas medidas de proximidade do ponto testado com o representante dos agrupamentos formados.

A rede neural utilizada para reconhecimento de padrões neste trabalho utiliza o método de treinamento supervisionado, e o modo de operação em batelada.

2.5. Vantagens e Desvantagens das RNA

As principais vantagens das RNA são:

- as redes neurais permitem análises superiores às conseguidas com técnicas estatísticas;

- o tempo necessário para implementar uma rede é normalmente menor que o utilizado para a construção de um sistema especialista equivalente;

- como as unidades da rede trabalham em paralelo, a destruição ou defeito em um de seus neurônios não torna a rede inoperante, não causando normalmente grandes problemas em seu funcionamento. Diz-se que a RNA é tolerante a falhas;

- as redes neurais têm sido utilizadas como filtro de dados devido à sua capacidade de separar ruídos dos dados relevantes;

- uma rede bem construída para uma determinada aplicação pode ser utilizada em tempo real, sem a necessidade de ter sua arquitetura alterada a cada atualização, bastando que seja "retreinada" com base nos novos dados que surgirem;

- as RNA possuem a habilidade de adaptação e aprendizagem;

- as RNA modelam mais facilmente sistemas não-lineares.

Algumas desvantagens das RNA são:

- não há sempre a garantia de bom desempenho e técnicas alternativas poderão ser melhores;

- o treinamento de uma rede, dependendo da aplicação, pode consumir muito tempo (várias horas);

- é impossível saber por que a rede chegou a uma determinada conclusão. Seus critérios decisórios são desconhecidos, não se sabendo quais pesos são relevantes para a tomada de decisão. Os milhares de pesos não aceitam interpretação e nem são passíveis de interpretação lógica: sabe-se apenas que funcionam;

- para que a rede possa aprender corretamente, necessita ser treinada com grande volume de dados;

- os dados de entrada necessitam de um tratamento prévio. Em alguns casos, devem ser normalizados e cuidadosamente selecionados para que a rede seja corretamente treinada. Dados de má qualidade produzem resultados falhos;

- não há regras bem estabelecidas e gerais para se determinar:

- o volume dos dados de entrada para treinamento;
- o número de unidades escondidas;
- o melhor método de treinamento;
- a percentagem dos dados que deve ser destinada ao treinamento e ao teste da rede.

2.6. Aplicações das RNA

As RNA são aproximadores universais. Reproduzem qualquer mapeamento contínuo em uma região de operação fechada e limitada, com o grau de precisão desejada, em espaços n-dimensionais. A Fig. 2.5 ilustra a aproximação de uma função realizada através de MATLAB®.

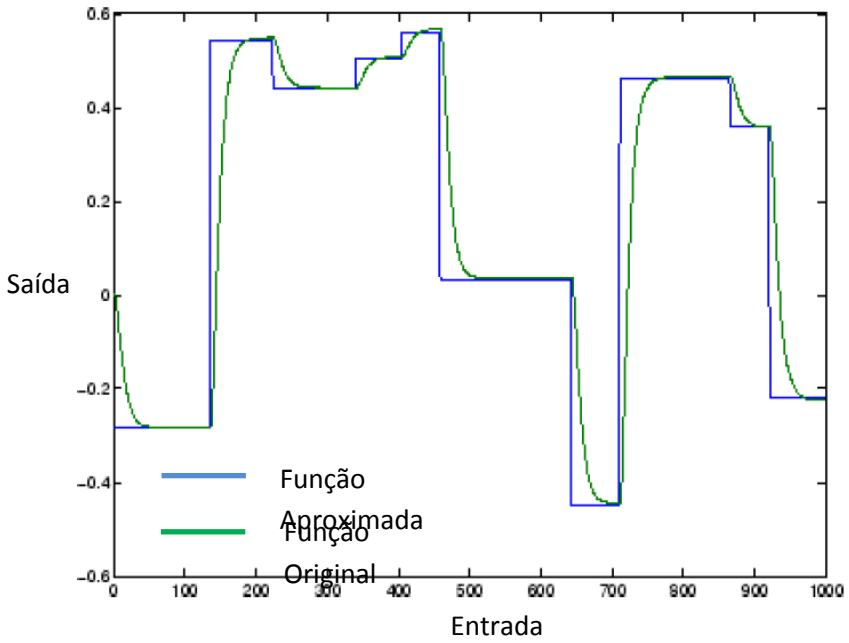


Fig.2.5. Exemplo de Aproximação de Função (MATLAB, 2009)

Algumas aplicações típicas são:

- realização de diagnósticos médicos;
- identificação de sistemas dinâmicos;
- identificação do modelo de um sistema não linear baseado nas amostras da série temporal;
- implementação de piloto automático de veículos e aeronaves e guiagem, de robô;
- visão computacional;
- modelagem de sistemas não lineares;
- controle de processo industrial;
- análise, síntese e reconhecimento de voz;
- operações matemáticas: SVD (*Singular Value Decomposition*), autovalores etc.;
- composição musical;
- neuroprótese.

As diversas aplicações podem ser divididas basicamente em cinco categorias: regressão, classificação, associação de dados, filtragem de dados e conceitualização de dados.

2.6.1. Aplicações em Regressão

As redes neurais são empregadas em regressão e no seu caso particular, predição.

Redes muito utilizadas para essa aplicação são as do tipo *Backpropagation,* ADALINE, e MADALINE.

Alguns exemplos de situações em que se utilizam RNA em regressão e predição são:

- calibração de instrumentos;
- aplicações financeiras: cotações da bolsa de valores etc.;
- abalos sísmicos;
- vazão de rios e represas;
- produtividade de culturas.

2.6.2. Aplicações em Classificação

As RNAs podem separar os valores de entrada em diferentes grupos, utilizando, *Backpropagation,* redes probabilísticas, LVQ (*Learning Vector Quantization)* e outras técnicas.

Alguns exemplos de aplicação são:

- reconhecimento de padrões de fala (palavras isoladas, frases ou fala contínua), de locutor (identificação e verificação) e de sons;
- reconhecimento de caracteres ópticos OCR (*Optical Character Recognizer*);
- reconhecimento de imagem: na indústria (controle da linha de montagem), em medicina (diagnóstico médico de mamografia, células cancerígenas, etc.), nas artes (detecção de autenticidade de quadros, em visão computacional (guiagem de robô) etc.;
- controle e otimização.

2.6.3. Aplicações em Associação de Dados

A utilização em associação de dados é semelhante à da classificação, porém, acrescida da possibilidade de classificar ou reconhecer dados ruidosos ou com erros.

Os tipos de rede geralmente utilizados para esse fim são Hopfield, Boltzman Machine, Redes Hamming, Memória Associativa Bidirecional (BAM), entre outros. Um exemplo típico é a detecção de caracteres incompletos ou ruidosos.

2.6.4. Aplicações em Filtragem de Dados

As redes neurais, quando utilizadas em filtragem de dados, normalmente realizam a filtragem adaptativa do sinal de entrada, treinados com algoritmos LMS (*Least Mean Squares*). As redes neurais podem ou não ser retroalimentadas.

Dois exemplos de filtragem por redes neurais são:

- cancelamento adaptativo de ruído;

- cancelamento do batimento cardíaco maternal em eletrocardiogramas do feto.

2.6.5. Aplicações em Conceitualização de Dados

As RNA podem analisar entradas (pontos no espaço de atributos) e agrupá-las segundo critérios de proximidade ou de correlação. Para tal, podem ser utilizados "mapas de auto-organização" (SOM, *Self-Organization Maps*) ou redes de ressonância adaptativa. Em ambos os casos o treinamento é não supervisionado.

Alguns exemplos de problemas solucionados por esses sistemas (MATLAB, 2009) são:

- problema do caixeiro viajante;

- extração em um banco de dados dos nomes dos prováveis compradores de um produto;

- agrupamento (*clustering*);

- rastreamento de alvo.

2.7. Generalização das RNA

Uma propriedade fundamental para o bom desempenho das RNA é a generalização (BRAGA; CARVALHO; LUDEMIR, 2010). Segundo esta propriedade, uma rede já treinada pode gerar respostas corretas mesmo que a entrada aplicada não tenha sido utilizada antes pelo sistema, desde que seja similar às que participaram do treinamento. Desse modo, ainda que o treinamento seja realizado com uma grande variedade de situações, a rede tende a gerar respostas corretas para todas as entradas.

Há situações em que após o treinamento o erro apresenta valor pequeno, porém os testes fornecem um valor de erro bem maior. Esse sobreajuste ocorre quando a rede apenas "memoriza os valores de entrada do treinamento", mas, devido à baixa variedade de situações não ocorre a necessária generalização.

Geralmente o sobreajuste pode ser resolvido na fase de planejamento da rede. Se a rede for suficientemente grande o problema será automaticamente resolvido, mas, nem sempre é simples saber qual o tamanho mínimo necessário para a RNA.

Existem duas ferramentas para evitar o sobreajuste: parada antecipada (*early stopping*) e regularização.

2.7.1. Parada Antecipada

A técnica de parada antecipada divide os dados disponíveis em três grupos: de treinamento, de validação e grupo de testes. O grupo de treinamento é utilizado para cálculo do gradiente e atualização dos pesos e polarizações.

Também é realizado o treinamento com o grupo de validação, mas dessa vez o objetivo é monitorar o erro. Durante o treinamento, o valor do erro tende a diminuir, mas quanto há sobreajuste, o valor do erro aumenta novamente.

Assim, durante o treinamento do grupo de validação, quando o erro começa a aumentar em um número predeterminado de iterações, o treinamento é interrompido e os valores dos pesos e polarizações voltam a ser aqueles obtidos quando o erro era o menor. O erro do grupo de testes é utilizado para verificar se os dados foram divididos de forma correta.

Durante a divisão dos dados para validação é importante escolher pontos representativos do grupo total de dados.

2.7.2. Regularização

Esta técnica consiste em alterar a função de processamento da RNA, utilizando a função de desempenho da rede F, apresentada em (2.3):

$$F = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 + (1-y)\frac{1}{n}\sum_{j=1}^{n}(W_j)^2$$

(2.3)

onde:

F: função de desempenho da rede;

N: número de iterações;

n: o número de entradas;

e_i: erro calculado de ordem i;

y: taxa de desempenho;

W_j: peso de ordem j.

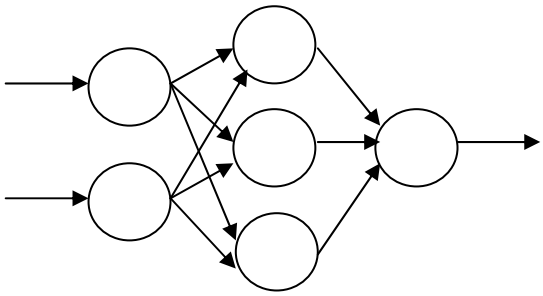
A maior dificuldade encontrada na utilização deste método é definir o valor ideal da taxa de desempenho **y**. Se o seu valor for muito pequeno, a rede poderá não ser eficiente e se for muito grande, há probabilidade de sobreajuste. A escolha adequada de **y** garante que o

treinamento gere pesos menores, resultando em um comportamento sujeito ao sobreajuste em menor grau.

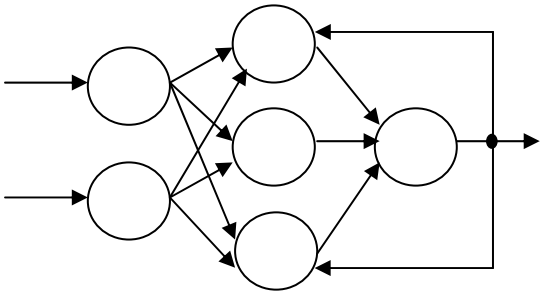
Ainda, para conseguir um valor otimizado da taxa de desempenho **y**, pode-se utilizar uma regularização bayesiana, uma inferência estatística que considera os pesos e polarizações como variáveis aleatórias e emprega métodos estatísticos para estimar os parâmetros de regularização (BEALE et al., 2010).

2.8. Arquitetura das Redes

As RNA podem variar quanto ao tipo de conexão entre os nodos, número de camadas, número de nodos por camadas e topologia. As redes podem ser alimentadas diretamente (Fig.2.6a) ou ser realimentadas (Fig.2.6b).



(a) Exemplo de RNA de Alimentação Direta



(b) Exemplo de RNA Realimentada

Fig. 2.6 Tipos de RNA quanto à Alimentação

As redes realimentadas, quando têm todas as saídas ligadas às entradas, são chamadas redes autoassociativas. Essas redes são úteis para a recuperação de padrão, pois associam um padrão de entrada a ele mesmo.

As RNA podem ser fracamente conectadas quando nem todos os neurônios se conectam entre si, ou fortemente conectadas quando há conexão entre todos os nodos. Podem ter apenas uma única camada ou serem multicamadas.

As redes de camada única possuem apenas um nodo entre qualquer entrada e qualquer saída da rede. Redes multicamadas, mostradas na Fig.2.7, possuem camadas intermediárias conhecidas como camadas escondidas (*hidden*), o que normalmente aumenta consideravelmente o seu número de aplicações, por exemplo, em reconhecimento de padrões. Existem algumas regras empíricas para definir o número ideal de camadas escondidas.

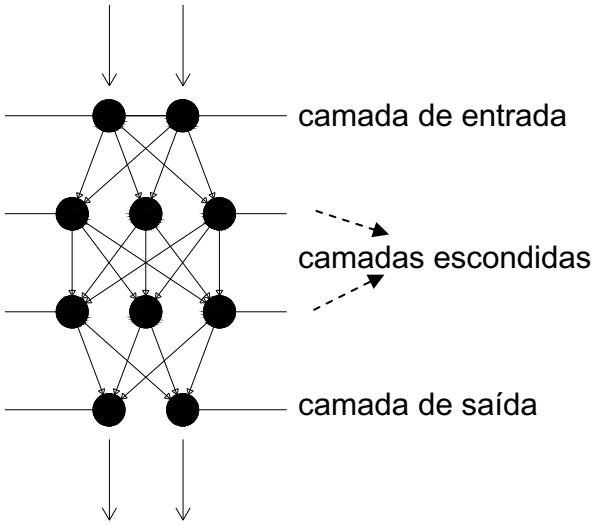


Fig.2.7. Exemplo de Rede Multicamadas

A arquitetura *Perceptron* Multicamadas (MLP, *Multi Layer Perceptron*) consiste de uma camada de entrada, uma ou mais camadas intermediárias (escondidas) e uma camada de saída. Pode utilizar funções de ativação "*hard-limiting*", sigmóide, tangente hiperbólica e outras. O algoritmo *Backpropagation* é geralmente utilizado para treinar o MLP, empregando o método do gradiente, que minimiza o erro médio quadrático entre a saída desejada (alvo) e a saída da rede.

2.9. Regras de Aprendizagem

As regras de aprendizagem podem basear-se em adaptação por correção de erro ou em algoritmos competitivos (SILVA; SPATTI; FLAUZINO, 2010).

- Aprendizagem adaptativa por correção de erro

Nesse tipo de aprendizagem é aplicado o vetor de entradas ao sistema e estabelecido o vetor de saídas esperadas para cada entrada (vetor de alvos D). Durante o treinamento, o sistema adapta seus pesos e polarizações de modo que o erro para a solução do problema fique abaixo do limiar estabelecido.

Redes lineares utilizam a *Regra Delta*, que utiliza a minimização dos erros quadráticos (LMS) entre a saída *desejada* e a saída *realmente fornecida* pela rede.

O algoritmo *Backpropagation*, descrito na subseção 2.10, utiliza uma ampliação dessa regra, conhecida como *Regra Delta Generalizada*.

- Aprendizagem por algoritmo competitivo

Essa regra de aprendizagem á aplicada em uma topologia em que os nodos de entrada são diretamente ligados aos nodos de saída que, por sua vez, podem possuir ligações laterais entre si.

Durante a aprendizagem, as saídas competem entre si para serem ativadas, de modo que a vencedora será ativada e terá seus pesos atualizados no treinamento. Os nodos de saída que possuem maior ativação inicial são os que têm maior probabilidade de ficarem mais fortes e inibirem os outros nodos ao longo do tempo.

2.10. Algoritmo *Backpropagation* (retropropagação)

A Fig.2.8 ilustra uma rede multicamadas em que é aplicado o algoritmo *Backpropagation* (SILVA; SPATTI; FLAUZINO, 2010), empregado neste trabalho.

O algoritmo compõe-se de duas fases: propagação para frente e para trás.

1ª fase: propagação para frente

Os valores de entrada são transformados nas camadas intermediárias até atingir a camada de saída da rede, onde o erro total final da 1ª fase é calculado. Os passos desta transformação são descritos a seguir.

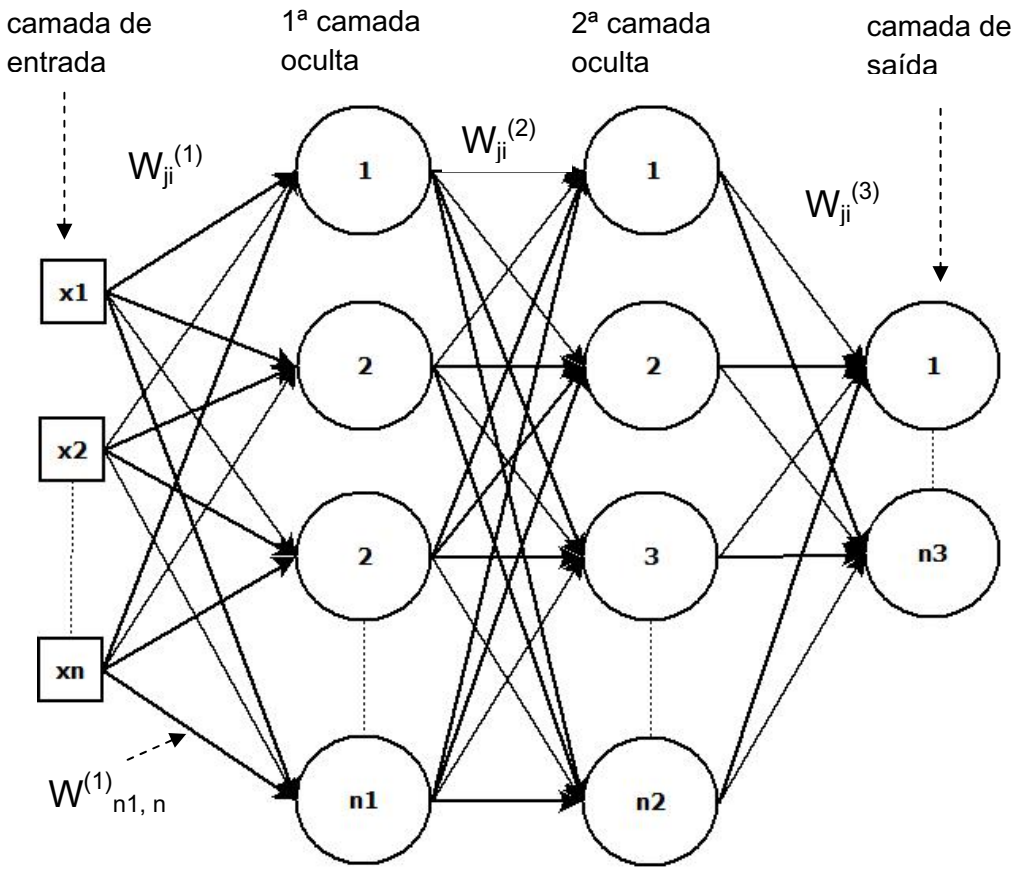


Fig. 2.8. Rede *Perceptron* Multicamada para Aplicação do Algoritmo *Backpropagation*

(1) O vetor $X = \{x_1, x_2, ..., x_n\}$ representa os valores de entrada da rede.

(2) As matrizes de pesos sinápticos entre as camadas são:

$W_{ji}^{(1)}$: entre a camada de entrada e a 1ª oculta; $j = 1, 2...n_1$; $i = 1, 2..n$.

$W_{ji}^{(2)}$: entre 1ª oculta e a 2ª oculta; $j = 1, 2...n_2$; $i = 1, 2..n_1$.

$W_{ji}^{(3)}$: entre a 2ª oculta e a de saída; $j = 1, 2...n_3$; $i = 1, 2..n_2$.

(3) Os vetores de entrada nas camadas $I_j^{(1)}$, $I_j^{(2)}$, $I_j^{(3)}$ são dados por:

$I_j^{(1)} = \sum_{i=0}^{n} W_{ji}^{(1)} . x_i$ para $j = 1,2 \ldots n_1$; entrada na 1ª camada oculta;

$I_j^{(2)} = \sum_{i=0}^{n_1} W_{ji}^{(2)} . Y_i^{(1)}$ para $j = 1,2 \ldots n_2$; entrada na 2ª camada oculta;

$I_j^{(3)} = \sum_{i=0}^{n_2} W_{ji}^{(3)} . Y_i^{(2)}$ para $j = 1,2 \ldots n_3$; entrada na camada de saída.

(4) Os vetores de saída das camadas $Y_j^{(1)}$, $Y_j^{(2)}$, $Y_j^{(3)}$ são:

$Y_j^{(1)} = g\left(I_j^{(1)}\right)$ j = 1, 2...n_1; saída da 1ª camada oculta;

$Y_j^{(2)} = g\left(I_j^{(2)}\right)$ j = 1, 2...n_2; saída da 2ª camada oculta;

$Y_j^{(3)} = g\left(I_j^{(3)}\right)$ j = 1, 2...n_3; saída da camada de saída.

sendo g a função de transferência nas camadas ocultas.

(5) O vetor de erro (função de erro) é calculado por:

$$E(k) = \frac{1}{2}\sum_{j=1}^{n3}\left(d_j(k) - Y_j^{(3)}(k)\right)^2$$

(2.4)

onde:

k : 1, 2,..., n_3;

E(k): vetor de erro ou função de erro (erro no nodo k);

$d_j(k)$: valor de saída desejado no nodo k;

$Y_j(k)$: valor de saída obtido no nodo k.

(6) O valor total final do erro na 1ª fase E_M é calculado pelo erro médio quadrático:

$$E_M = \frac{1}{n3}\sum_{k=1}^{n3} E(k)$$

(2.5)

onde:

n_3: número de nodos da saída;

E(k): vetor de erro ou função de erro (erro no nodo k), dada em (2.4).

2ª fase: propagação para trás (*backpropagation*)

Os valores dos pesos são ajustados desde a saída da rede até a sua entrada, em função dos erros calculados. A otimização pelo gradiente do erro em relação aos pesos é realizada camada por camada, começando pelos pesos sinápticos que ligam a segunda camada intermediária à saída.

(7) Aplicação do gradiente (regra da cadeia):

$$\nabla E^{(3)} = \frac{\partial E}{\partial W_{ji}^{(3)}} = \frac{\partial E}{\partial Y_j^{(3)}} \cdot \frac{\partial Y_j^{(3)}}{\partial I_j^{(3)}} \cdot \frac{\partial I_j^{(3)}}{\partial W_{ji}^{(3)}}$$

(2.6)

onde;

E: vetor de erro ou função de erro, dada em (2.4);

$W_{ji}^{(3)}$: matriz de pesos entre a 2ª camada oculta e a de saída; j = 1, 2...n_3 ; i = 1, 2..n_2;

$\mathbf{Y_j^{(3)}}$ e $\mathbf{I_j^{(3)}}$: dados nos passos (4) e (3) do algorítmo, respectivamente.

Sendo:

$$\frac{\partial E}{\partial Y_j^{(3)}} = -\left(d_j - Y_j^{(3)}\right)$$

(2.7a)

$$\frac{\partial Y_j^{(3)}}{\partial I_j^{(3)}} = g'\left(I_j^{(3)}\right)$$

(2.7b)

$$\frac{\partial I_j^{(3)}}{\partial W_{ji}^{(3)}} = Y_i^{(2)}$$

(2.7c)

onde:

g: função de transferência nas camadas ocultas.

Com (2.7) em (2.6), resulta:

$$\frac{\partial E}{\partial W_{ji}^{(3)}} = -\left(d_j - Y_j^{(3)}\right).g'\left(I_j^{(3)}\right).Y_i^{(2)}$$

(2.8)

(8) A variação dos pesos deve ter sinal contrário ao do gradiente (SILVA, et al., 2010).

$$\Delta W_{ji}^{(3)} = -\eta . \frac{\partial E}{\partial W_{ji}^{(3)}}$$

(2.9)

A equação (2.8) pode ser reescrita como:

$$\Delta W_{ji}^{(3)} = \eta . \delta_j^{(3)} . Y_i^{(2)}$$

(2.10)

onde:

$\delta_j^{(3)} = \left(d_j - Y_j^{(3)}\right).g'\left(I_j^{(3)}\right)$;

η: taxa de aprendizagem .

Assim, o novo valor do peso será:

$$W_{ji}^{(3)}(t+1) = W_{ji}^{(3)}(t) + \eta . \delta_j^{(3)} . Y_i^{(2)}$$

(2.11)

onde:

$W_{ji}^{(3)}$: matriz de pesos entre a 2ª camada oculta e a de saída; j = 1, 2...n_3 ; i = 1, 2..n_2;

$\mathbf{Y^{(2)}_j}$: saída da 2ª camada oculta, dada no passo (4);

$\boldsymbol{\delta^{(3)}}\mathbf{j}$: dada em (2.10);

t: iteração (*epoch*).

(9) O mesmo processo repete-se para os pesos entre as outras camadas. Assim, para a otimização dos pesos entre a 1ª e 2ª camadas ocultas, tem-se:

$$\nabla E^{(2)} = \frac{\partial E}{\partial W_{ji}^{(2)}} = \frac{\partial E}{\partial Y_j^{(2)}} . \frac{\partial Y_j^{(2)}}{\partial I_j^{(2)}} . \frac{\partial I_j^{(2)}}{\partial W_{ji}^{(2)}}$$

(2.12)

sendo:

$$\frac{\partial Y_j^{(2)}}{\partial I_j^{(2)}} = g'\left(I_j^{(2)}\right)$$

(2.13a)

$$\frac{\partial I_j^{(2)}}{\partial W_{ji}^{(2)}} = Y_j^{(1)}$$

(2.13b)

$$\frac{\partial E}{\partial Y_j^{(2)}} = \sum_{k=1}^{n3} \frac{\partial E}{\partial I_k^{(3)}} \cdot \frac{\partial I_k^{(3)}}{\partial Y_j^{(2)}} = \sum_{k=1}^{n3} \frac{\partial E}{\partial I_k^{(3)}} \cdot \frac{\partial \left(\sum_{k=1}^{n2} W_{kj}^{(3)} \cdot Y_j^{(2)}\right)}{\partial Y_j^{(2)}}$$

(2.13c)

$$\frac{\partial E}{\partial Y_j^{(2)}} = \sum_{k=1}^{n3} \frac{\partial E}{\partial I_k^{(3)}} \cdot W_{kj}^{(3)}$$

(2.13d)

$$\frac{\partial E}{\partial Y_j^{(2)}} = -\sum_{k=1}^{n3} \delta_k^{(3)} \cdot W_{kj}^{(3)}$$

(2.13e)

Com (2.13 a, b, e) em (2.12), resulta:

$$\frac{\partial E}{\partial W_{ji}^{(2)}} = -\left(\sum_{k=1}^{n3} \delta_k^{(3)} \cdot W_{kj}^{(3)}\right) \cdot g'\left(I_j^{(2)}\right) \cdot Y_i^{(1)}$$

(2.14)

(10) A variação dos pesos é então:

$$\Delta W_{ji}^{(2)} = -\eta \cdot \frac{\partial E}{\partial W_{ji}^{(2)}}$$

(2.15)

A equação (2.15) pode ser reescrita como:

$$\Delta W_{ji}^{(2)} = \eta \cdot \delta_j^{(2)} \cdot Y_i^{(1)}$$

(2.16)

sendo:

$$\delta_j^{(2)} = \left(\sum_{k=1}^{n3} \delta_k^{(3)} . W_{kj}^{(3)}\right) . g'\left(I_j^{(2)}\right)$$

(2.17)

Assim, o novo valor do peso será:

$$W_{ji}^{(2)}(t+1) = W_{ji}^{(2)}(t) + \eta . \delta_j^{(2)} . Y_i^{(1)}$$

(2.18)

onde:

$W_{ji}^{(2)}$: pesos entre 1ª e a 2ª camadas ocultas;

$\mathbf{Y^{(2)}_j}$: saída da 2ª camada oculta, dada no passo (4);

$\boldsymbol{\delta^{(2)}j}$: dada em (2.17);

t: iteração (*epoch*).

(11) Finalmente, de modo análogo, para a otimização dos pesos entre a camada de entrada e a 1ª camada oculta, tem-se;

$$\nabla E^{(1)} = \frac{\partial E}{\partial W_{ji}^{(1)}} = \frac{\partial E}{\partial Y_j^{(1)}} . \frac{\partial Y_j^{(1)}}{\partial I_j^{(1)}} . \frac{\partial I_j^{(1)}}{\partial W_{ji}^{(1)}}$$

(2.19)

sendo:

$$\frac{\partial I_j^{(1)}}{\partial W_{ji}^{(1)}} = xi$$

(2.20a)

$$\frac{\partial Y_j^{(1)}}{\partial I_j^{(1)}} = g'\left(I_j^{(1)}\right)$$

(2.20b)

$$\frac{\partial E}{\partial Y_j^{(1)}} - \sum_{k=1}^{n2} \frac{\partial E}{\partial I_k^{(2)}} . \frac{\partial I_k^{(2)}}{\partial Y_j^{(1)}} - \sum_{k=1}^{n2} \frac{\partial E}{\partial I_k^{(2)}} . \frac{\partial\left(\sum_{k=1}^{n1} W_{kj}^{(2)} . Y_j^{(1)}\right)}{\partial Y_j^{(1)}}$$

(2.20c)

$$\frac{\partial E}{\partial Y_j^{(1)}} - \sum_{k=1}^{n2} \frac{\partial E}{\partial I_k^{(2)}} . W_{kj}^{(2)}$$

(2.20d)

$$\frac{\partial E}{\partial Y_j^{(1)}} = -\sum_{k=1}^{n2} \delta_k^{(2)} . W_{kj}^{(2)}$$

(2.20d)

Com (2.20 a,b,d) em (2.19), resulta:

$$\frac{\partial E}{\partial w_{ji}^{(1)}} = -\left(\sum_{k=1}^{n2} \delta_k^{(2)} . w_{kj}^{(2)}\right) . g'\left(I_j^{(1)}\right) . xi$$

(2.21)

(12) Variação dos pesos:

$$\Delta w_{ji}^{(1)} = -\eta . \frac{\partial E}{\partial w_{ji}^{(1)}}$$

(2.22)

Assim:

$$\Delta w_{ji}^{(1)} = \eta . \delta_j^{(1)} . xi$$

(2.23)

sendo:

$$\delta_j^{(1)} = \left(\sum_{k=1}^{n2} \delta_k^{(2)} . w_{kj}^{(2)}\right) . g'\left(I_j^{(1)}\right)$$

(2.24)

Finalmente:

$$w_{ji}^{(1)}(t+1) = w_{ji}^{(1)}(t) + \eta . \delta_j^{(1)} . xi$$

(2.25)

Os cálculos indicados pelas equações do algoritmo são efetuados sucessivamente até que se atinja um erro mínimo especificado ou até que se chegue a um número máximo de interações (*epochs*) pré-determinado. A Fig.2.9 e a Fig.2.10 apresentam o fluxograma para essas duas situações de parada do algoritmo.

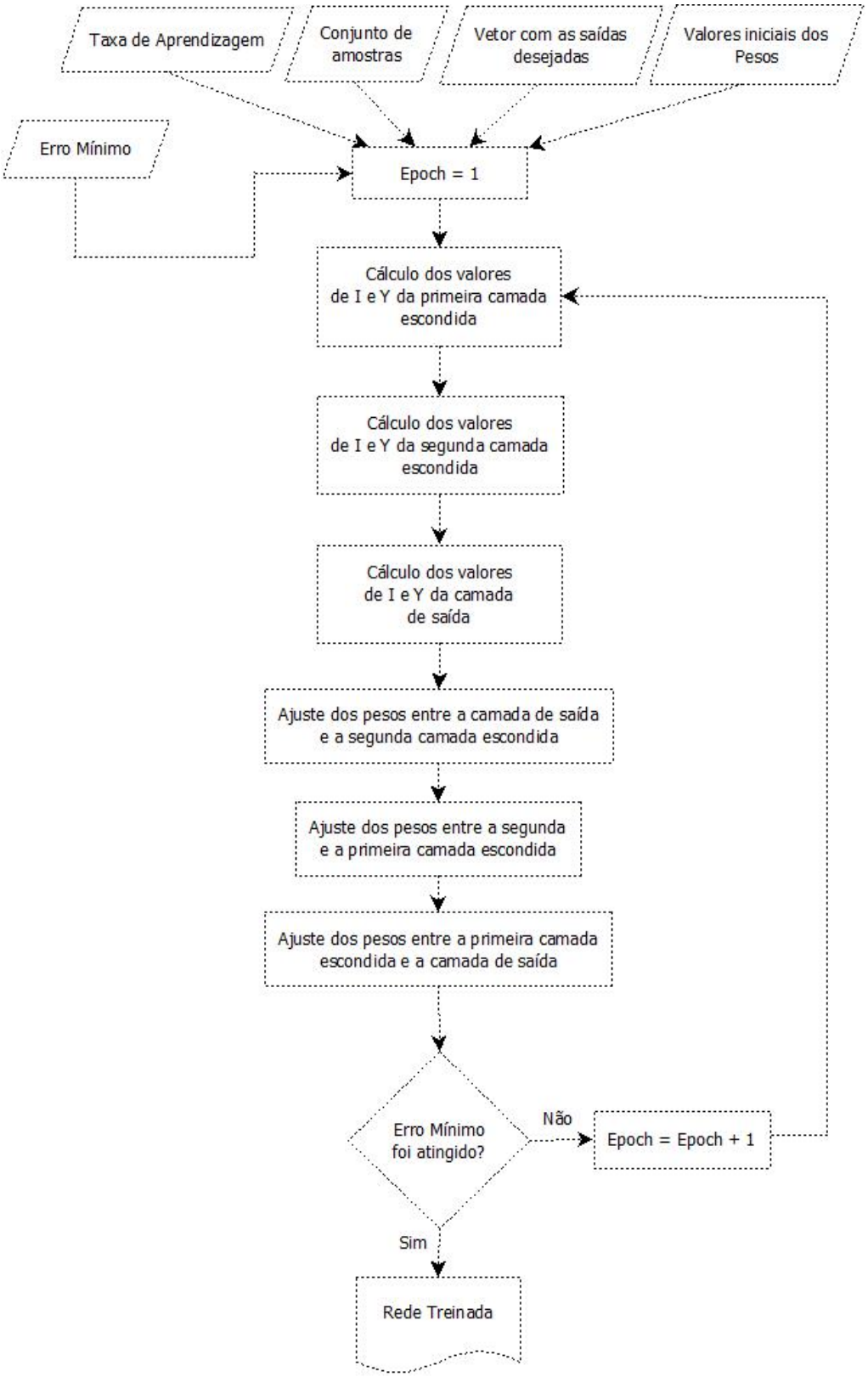


Fig. 2.9. Treinamento com *Backpropagation* Controlado por Erro Mínimo Especificado

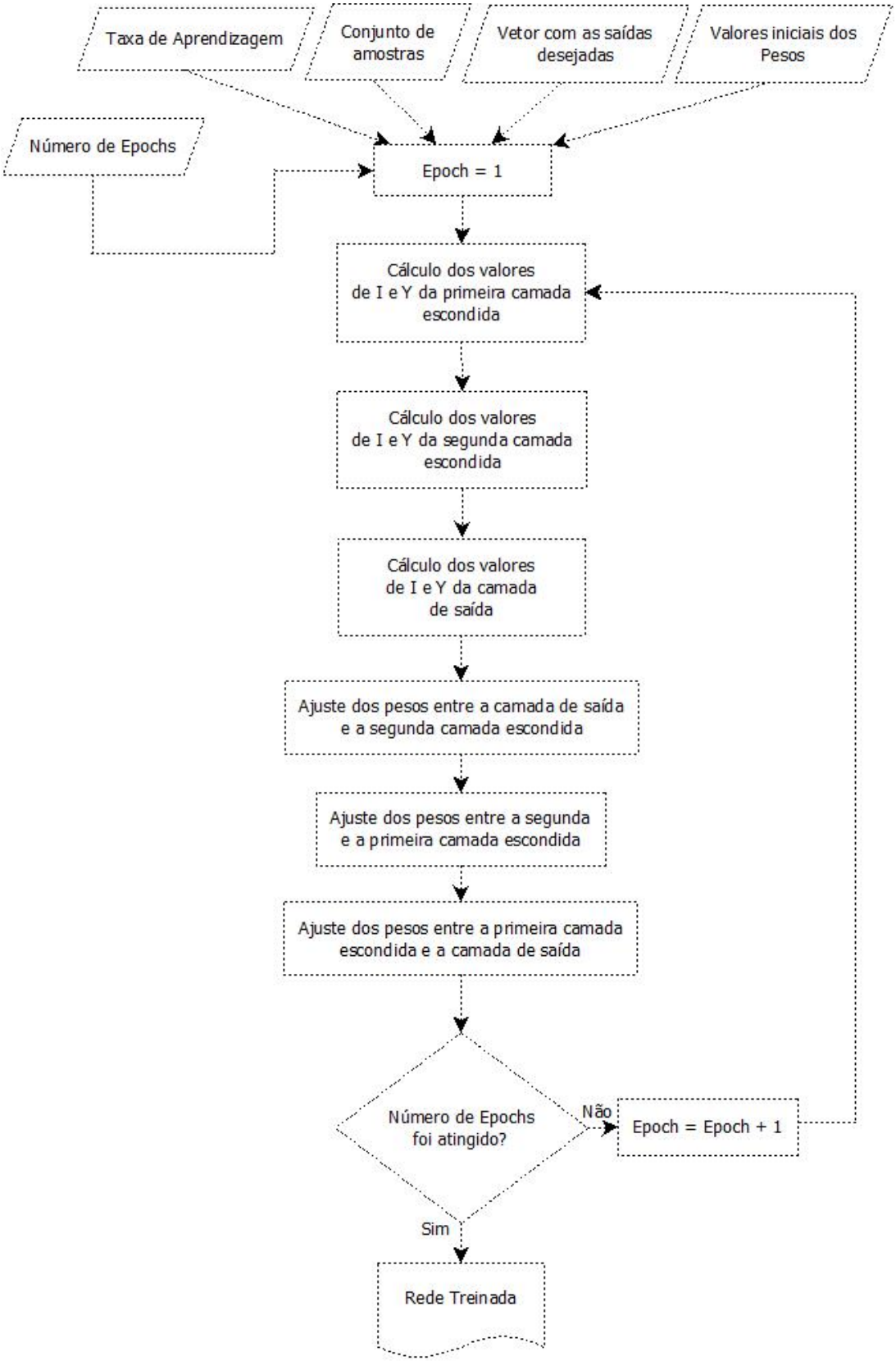


Fig.2.10. Treinamento com *Backpropagation* Controlado pelo Número de *Epochs*

2.11. Outros Tipos de Redes

Outros tipos de redes neurais para aplicações específicas são a rede de base radial, a rede probabilística, as redes auto-organizáveis, as redes recorrentes etc.

- Rede de base radial

Redes neurais de base radial possuem alimentação direta e apresentam três camadas: uma de entrada, uma oculta com função de ativação gaussiana e alta dimensionalidade e uma camada de saída linear.

A alta dimensionalidade procura atender ao teorema de Cover (1965), que enuncia que um problema de classificação de padrões de alta dimensionalidade tem maior probabilidade de ser linearmente separável do que um de baixa dimensionalidade. Quanto maior a dimensão do espaço oculto da rede, maior será a precisão do resultado (HAYKIN, 1999).

- Rede probabilística

As redes probabilísticas são utilizadas para solucionar problemas de classificação. Ao receber um vetor de entrada de teste **T**, a primeira camada da rede gera outro vetor **P** cujos elementos representam a distância (proximidade) entre os valores de **T** e os valores da entrada de treinamento **Tr**. A segunda camada realiza a soma desses valores, para cada classe de entrada, produzindo um vetor de saída com os valores das probabilidades de pertinência à classe C. Na saída da segunda camada há uma função de transferência que, a partir da maior probabilidade, define o valor 1 para essa classe C e o valor 0 para as demais classes.

- Redes auto-organizáveis

Essas redes, também conhecidas como SOM (*Self Organizing Maps*) ou redes de Kohonen, possuem a propriedade de modificar e otimizar seus parâmetros sem ajuda do meio externo, ou seja, seu aprendizado é não supervisionado. Utilizam algoritmo competitivo para solucionar problemas, geralmente de classificação ou de agrupamento de dados. Basicamente possuem apenas duas camadas, entrada e saída, porém podem ter camadas intermediárias, dependendo da aplicação. As SOM´s baseiam-se no córtex cerebral, o qual ativa uma região específica, dependendo da natureza dos estímulos (visuais, auditivos etc).

- Redes Recorrentes de Hoppfield

São redes de uma única camada realimentada, com todos os neurônios interligados entre si. Suas principais características são o comportamento dinâmico, a possibilidade de memorização de relacionamentos e de armazenamento informações.

As redes recorrentes podem ser facilmente implementadas em hardware analógico.

CAPÍTULO 3 - PROCESSAMENTO DE SINAIS DE FALA PARA O SEU RECONHECIMENTO

3.1. Introdução

A fala é uma forma de comunicação em que um locutor transmite uma mensagem a um ouvinte. Desse modo, existem dois processos de igual importância nessa comunicação: a produção da fala e a percepção, ou reconhecimento, da fala. (HAYKIN, 1999).

Alguns estudos sobre processamento de fala são voltados à teoria da informação, ou seja, referem-se à mensagem transmitida pela fala e seu significado. Outros estudos dão enfoque ao sinal que carrega a mensagem, trabalhando as suas características acústicas, por meio de métodos de codificação e parametrização. Assim, para um sistema de comunicação por fala apresentar um bom desempenho, deve obedecer às seguintes condições gerais (RABINER; SCHAFER, 1978):

- o sinal de fala deve ser representado e processado adequadamente, de modo que possa ser facilmente transmitido ou armazenado;

- o sinal de fala recebido deve ser uma cópia aceitável do que era em sua origem.

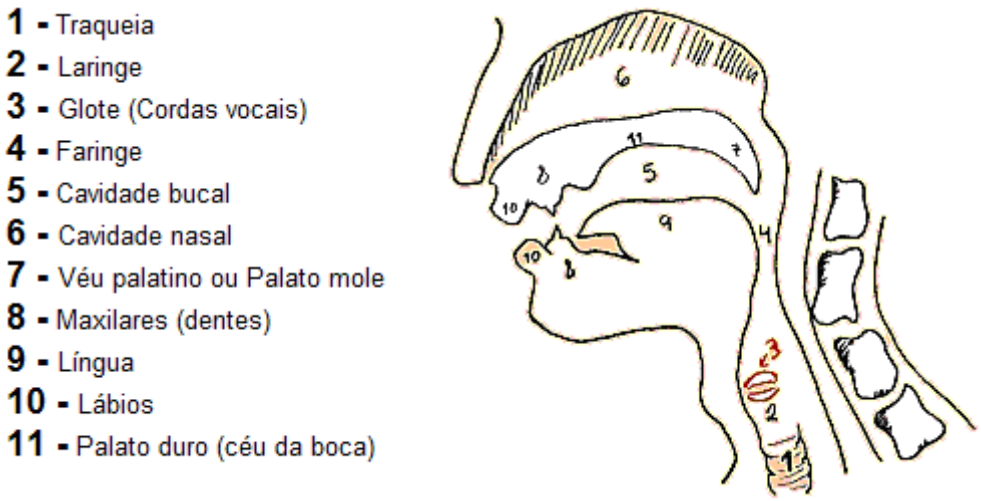
3.2. Produção da Fala

A produção da fala tem início no cérebro do locutor, onde é formulada a informação que posteriomente será transmitida. No passo seguinte, a informação é codificada, ou seja, é transformada em uma sequência de fonemas que representem as palavras do idioma, o qual deverá ser conhecido pelo ouvinte para que haja comunicação. São as ações neuromusculares do sistema vocal humano que convertem a informação em energia acústica. Resumindo, a mensagem é a manifestação física da informação. (RABINER; SCHAFER, 1978):

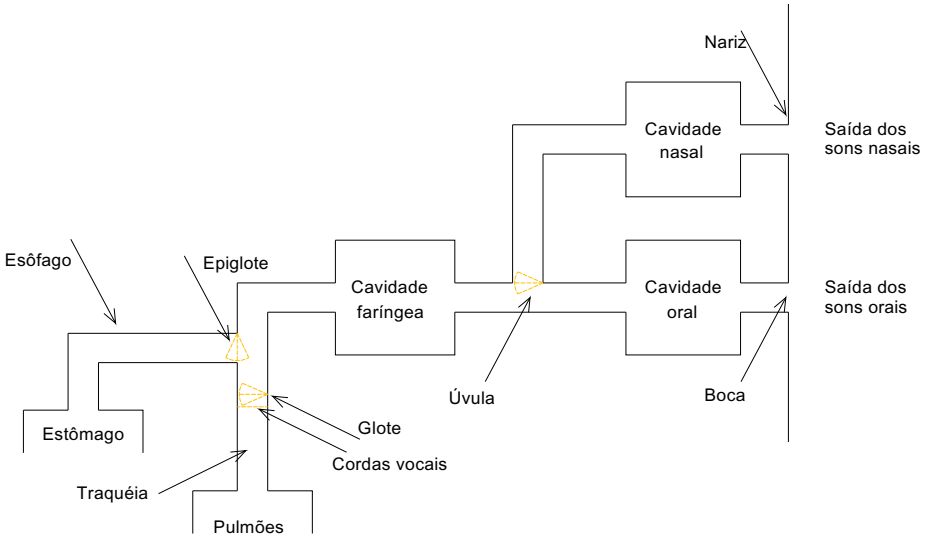
3.2.1. Sistema Vocal Humano

O esquema do aparelho fonador humano está mostrado na Fig.3.1. A produção da voz é iniciada com o ar expelido pelos pulmões.

O ar passa então pelo trato vocal, formado pela laringe, faringe e cavidade bucal. No caso de fonemas nasais, passa também pelo trato nasal.



(a) Sistema vocal humano



(b) Representação esquemática

Fig. 3.1. Aparelho Fonador Humano

3.2.1 O Trato Vocal e o Nasal

O trato vocal consiste de um tubo acústico com perdas, cuja seção reta varia com o tempo e ao longo do seu comprimento (RABINER; SCHAFER, 1978). É formado pela glote, onde estão as dobras ou "cordas" vocais, a faringe e cavidade oral (boca). Sua área transversal varia entre 0 e 20 cm², dependendo, do posicionamento da língua, lábios, mandíbulas e véu palatino. O posicionamento desses articuladores modifica as características do som a ser produzido.

O trato nasal tem início no véu palatino e termina nas fossas nasais. O véu palatino funciona como uma válvula que, ao ser aberta, acopla o trato nasal ao trato vocal, criando assim os sons nasais.

Os sons vozeados são criados pela excitação do trato (vocal ou nasal) por pulsos quase periódicos de ar causados pela vibração das dobras vocais.

Os sons não vozeados são formados pela simples aproximação das dobras vocais, que nesse caso não vibram.

Sons fricativos são gerados pela passagem forçada do ar através de uma constrição criada pelos articuladores, produzindo assim uma turbulência com efeito semelhante ao ruído.

Os sons explosivos, ou oclusivos, são formados pela súbita expansão do ar causada pelo aumento da pressão devido ao fechamento total do trato vocal, seguida de uma abertura abrupta. (RABINER; JUANG, 1993).

3.3. Reconhecimento da Fala

O reconhecimento de fala é um dos exemplos clássicos de reconhecimento de padrões, razão pela qual os seus fundamentos serão apresentados a seguir.

3.3.1. Reconhecimento de Padrões

Pode-se definir padrão como um conjunto de características ou propriedades que permitem o agrupamento de determinados objetos semelhantes, contidos em uma determinada classe ou categoria, tendo em vista a interpretação de dados de entrada que possibilitem a extração de características essenciais destes objetos.

Os grupos formados por objetos que possuem os mesmos valores para atributos pré-determinados constituem as classes.

Reconhecimento de padrões é uma tarefa que visa a classificar dados em diferentes classes, com base em algumas de suas características.

O processo de reconhecimento inicia-se na aquisição dos dados de entrada. A seguir são extraídas as características a serem comparadas, e por fim é realizada uma comparação entre as características extraídas e as características esperadas em cada classe. O objeto será designado à classe em que ele melhor se enquadra (CASTRO, 2001).

O processo de reconhecimento de padrões compreende três fases principais: aquisição, redução da dimensão e classificação.

A primeira fase é a aquisição ou medição do sinal a ser analisado. O sinal adquirido no meio físico, por um sensor, é representado por um vetor padrão de amostragem composto de **N** características: $X(i) = (X_1, X_2, X_3, ..., X_N)$.

Na segunda fase do reconhecimento, um seletor reduz a dimensão N do vetor de características, dele extraindo apenas as características principais mais significativas, as chamadas "*features*", para que o reconhecimento possa ser realizado com um menor tempo de processamento. A escolha das características a serem analisadas é de grande importância para a eficiência do sistema. Estas características variam de acordo com o tipo de sinal e o tipo de classificação a ser utilizado.

Obtido o vetor reduzido de características, o sistema inicia sua terceira fase, a *classificação* baseada na comparação entre as características principais ou fundamentais (*features*) e as representantes da classe.

Os classificadores podem ser paramétricos ou não paramétricos (TOU, 1981).

- classificador paramétrico: no caso do treinamento do classificador exigir amplo conhecimento inicial da estrutura estatística dos padrões a serem analisados e o padrão de entrada for identificado como pertencente a uma classe predefinida pelos padrões de treinamento. A classificação se processa de forma supervisionada.
- classificador não paramétrico: no caso do classificador utilizar um modelo estatístico que vai se ajustando progressivamente diante de processos adaptativos e a associação entre os padrões é baseada em similaridades entre os padrões de treinamento. A classificação se processa de forma não supervisionada.

Na execução de um projeto de reconhecimento de padrões, a grande dificuldade está na escolha da técnica adequada para que as fases de reconhecimento representem satisfatoriamente a realidade desejada.

- Técnicas empregadas na classificação

Algumas técnicas utilizadas para a classificação de padrões são as distâncias (euclidiana, de Hamming, de Mahalanobis), quantização vetorial (VQ, Vector Quantization), os modelos ocultos de Markov (HMM, *Hidden Markov Models*), as RNA (Redes Neurais Artificiais) e suas variações.

3.3.2. Fases do Reconhecimento de Fala

A percepção, ou reconhecimento da fala, é o processo inverso da produção. O reconhecimento tem como objetivo capturar uma mensagem acústica de fala, no meio físico, e interpretar a informação nele contida.

Para a execução dessa tarefa, a primeira fase é a aquisição do sinal através dos movimentos da membrana basilar do sistema auditivo do ouvinte. A mensagem adquirida é

processada e passa por uma transdução neural, que representa a extração das suas características, por meio da ativação do nervo auditivo.

Finalmente, o cérebro interpreta a mensagem transformada pelo sistema auditivo para reproduzir a informação.

Um sistema digital de comunicação por fala funciona de forma análoga à comunicação humana. Em um sistema texto-fala (TTS, *Text - to - Speech*), por exemplo, o texto digitado por um usuário é a mensagem. Ela é digitalizada, processada pelo sintetizador e convertida na fala correspondente ao texto.

Em um sistema digital para reconhecimento da fala, mostrado na Fig.3.2 e desenvolvido nesta dissertação, o processo é iniciado com a aquisição da mensagem de voz pelo microfone, o transdutor que a converte em um sinal de fala (tensão ou corrente elétrica variável com o tempo). O sinal é digitalizado e dele são extraídas as suas características principais (*features)*.

As *features* do sinal adquirido são comparadas com as dos sinais de fala previamente armazenadas, possibilitando à maquina tomar uma decisão.

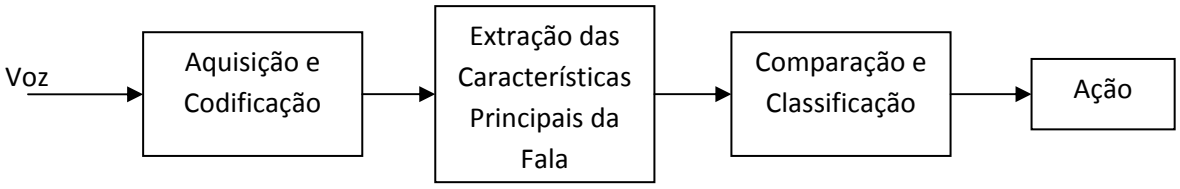


Fig.3.2. Diagrama de um sistema de reconhecimento da fala

3.3.3 Pré-processamento da Fala para o Reconhecimento

O pré-processamento do sinal de fala consiste em: amostragem, quantização, codificação, normalização, segmentação e janelamento, conforme ilustra a Fig.3.3.

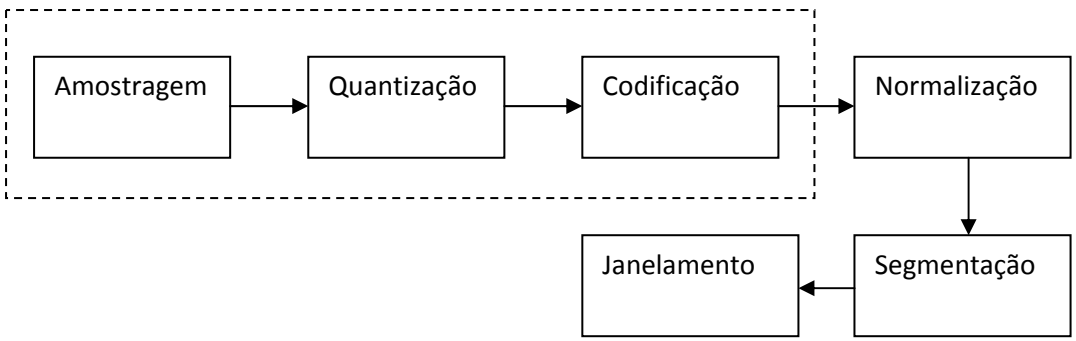


Fig.3.3 Fases do Pré-processamento

- Amostragem, quantização e codificação

As fases de amostragem, quantização e codificação são efetuadas automaticamente por um software escolhido, em que são especificados o tipo de arquivo de armazenamento e a taxa de amostragem do sinal de voz.

- Normalização (ou escala)

Normalização é o processo de escala da amplitude do sinal dos elementos do vetor X(i) que representa o sinal de fala, para manter seus valores entre limites convenientes, por exemplo, entre 1 e -1.

A normalização pode ser relativa ou absoluta.

A normalização absoluta consiste em dividir o vetor X(i) pelo maior valor decimal utilizado na fase de codificação, conforme expressão (3.1).

$$X_{norm}(i) = X(i) / 2^n \qquad i = 1, 2, 3...N \tag{3.1}$$

onde:

N: dimensão do vetor [inteiro positivo];
n: número de bits da codificação [inteiro positivo];
X(i): vetor de dimensão N que representa o sinal de fala, i= 1,2, 3...N;
$X_{norm}(i)$: vetor normalizado absolutamente.

A normalização relativa é definida por:

$$X(i) / Max(| X(i)|) \qquad i =1,2,...N \tag{3.2}$$

onde:

X(i): vetor de dimensão N que representa o sinal de fala, i= 1, 2, 3, ..., N;
Max |(X)|: máximo valor do módulo de X.

- Segmentação do sinal por janelas

Para o processamento de fala, a capacidade dos processadores de atuar em algumas funções e características de curto prazo exige que o vetor de fala seja dividido em pequenas partes de duração definida.

A duração Ts destes segmentos é escolhida de acordo com o período em que o sinal pode ser considerado estacionário. Os órgãos e músculos que compõem o aparelho fonador

movimentam-se lentamente, de modo que o sinal de fala geralmente apresenta comportamento estacionário entre cerca de 20 ms a 30 ms.

Uma desvantagem da segmentação seria a possibilidade de que a divisão de uma característica da fala em dois segmentos dificultasse então a sua posterior análise. Para evitar esse problema, utiliza-se a técnica da sobreposição do fim de um segmento com o início do próximo.

A sobreposição é expressa pela percentagem do segmento (final) que também pertence ao segmento seguinte (início). Os valores mais utilizados são de 25% e de 50% (McLOUGHLIN, 2009).

A segmentação é, portanto, a operação de multiplicar intervalos sucessivos do inteiro sinal no domínio do tempo, x(n), por um pulso retangular, w(n), de duração igual a Ts =N Ta, sendo N o número de amostras e Ta o período de amostragem. Esse pulso w(n) é denominado janela.

Uma vez que a multiplicação no domínio do tempo equivale à convolução no domínio da frequência, a janela retangular produz uma distorção do sinal no domínio da frequência, que deve estar dentro dos limites toleráveis da aplicação visada.

Assim, para suavizar a sobreposição e evitar efeitos de distorções intoleráveis, são utilizadas janelas especiais para realizar a segmentação, tais como, as de Hanning (cosseno), Hamming, Kaiser-Bessel , Blackman etc.

3.3.4. Características Principais da Fala Utilizadas no seu Reconhecimento

Há várias características principais que podem ser utilizadas no reconhecimento de fala (RABINER; JUANG, 1993).

Serão apresentadas, a seguir, as *features* utilizadas ou experimentadas no projeto de reconhecimento de fala desenvolvido neste trabalho: LPC (*Linear Predictive Coding*), Taxa de Cruzamento de Zeros, Energia de Curto Prazo e Coeficientes Mel-Cepstral.

3.3.4.1. Codificação Preditiva Linear (LPC, Linear Predictive Coding)

LPC é um dos modelos mais utilizados para representação do trato vocal. Suas vantagens são: possibilidade de gerar um modelo bem aproximado do sinal de fala, base matemática do modelo, simplicidade de implementação, tanto em software quanto em hardware, e um bom poder de discriminação no reconhecimento de fala e de locutor (THIANG; WIJOYO, 2011).

O LPC pressupõe que uma amostra de uma locução pode ser prevista pela combinação linear das amostras anteriores, por meio da minimização da soma da diferença dos quadrados entre as amostras originais e as linearmente previstas.

$$s(n) \cong a_1 s(n-1) + a_2 s(n-2) + ... + a_3 s(n-3) + ... + a_p s(n-p) \tag{3.3}$$

onde:

s(n): amostra de ordem n do sinal de voz;

p: ordem da LPC;

a_p: coeficiente LPC de ordem p.

Adicionando-se um termo que represente a excitação no sistema, obtém-se:

$$s(n) = \sum_{i=1}^{p} a_i s(n-i) + Gu(n) \tag{3.4}$$

onde:

G: ganho da excitação;

u(n): excitação normalizada.

Aplicando a Transformada Z à equação (3.4), obtém-se:

$$S(z) = \sum_{i=1}^{p} a_i \, z^{-1} S(z) + GU(z) \tag{3.5}$$

Por (3.5), chega-se à função de transferência H(z):

$$H(z) = \frac{S(z)}{GU(z)} = \frac{1}{1 - \sum_{i=1}^{p} a_i \, z^{-1}} = \frac{1}{A(z)} \tag{3.6}$$

Analisando a função (3.6), depreende-se que um sinal de fala pode ser representado pela saída de um sistema H(z), excitado por uma fonte normalizada u(n) com ganho G, conforme ilustra a Fig. 3.4.

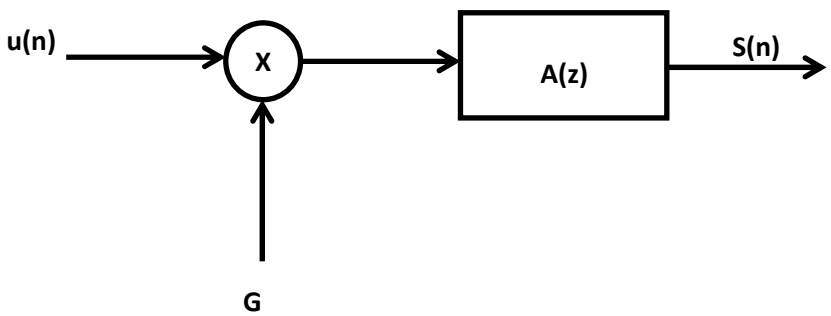


Fig.3.4. Modelo Simplificado do LPC

O sinal de fala pode apresentar características diferentes, caso seja vozeado ou não vozeado.

O sinal de fala vozeado é representado por um trem de pulsos quase periódico, enquanto o sinal não vozeado é representado por uma fonte de ruídos.

Com base nesses dois tipos de sinal de fala, é acrescida uma chave ao modelo LPC, para selecionar o gerador de trem de impulsos ou gerador randômico de ruídos. O sinal **u(n)** é amplificado com um ganho **G** e em seguida entra no filtro digital, que representa o trato vocal do locutor, conforme mostra a Fig.3.5.

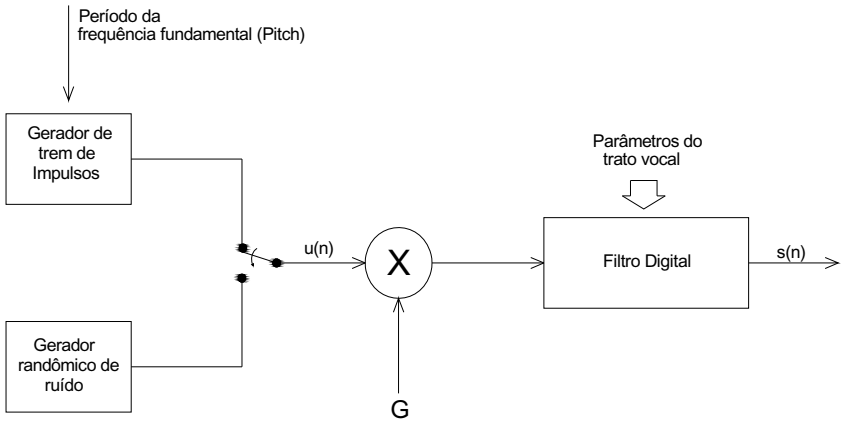


Figura 3.5. Esquema Completo do Modelo LPC

Considerando $\tilde{s}(n)$ a combinação linear das amostras anteriores do sinal de fala, resulta:

$$\tilde{s}(n) = \sum_{k=1}^{p} a_k s(n-k) \tag{3.7}$$

O erro do LPC é dado por:

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \tag{3.8}$$

Assim, a função de transferência do sistema é expressa por:

$$A(Z)=\frac{E(Z)}{S(Z)}=1-\sum_{k=1}^{p}a_k Z^{-k} \tag{3.9}$$

O principal desafio do sistema é determinar os coeficientes do LPC de modo que as características do filtro digital se aproximem ao máximo do sinal original (KATAGIRI, 2000).

Com a segmentação e janelamento resultam:

$$s_n(m)=s(n+m) \tag{3.10}$$

$$e_n(m)=e(n+m) \tag{3.11}$$

Com (3.10) e (3.11) em (3.8), obtém-se erro quadrático E_n:

$$E_n=\sum_m\left[s_n(m)-\sum_{k=1}^{p}a_k s_n(m-k)\right]^2 \tag{3.12}$$

O mínimo da função de erro E_n é obtido nos pontos em que a derivada parcial com relação a cada coeficiente LPC é nula (teoria da minimização).

$$\frac{\partial E_n}{\partial a_k}=0 \qquad \text{k =1, 2, ..., p} \tag{3.13}$$

Assim, resulta:

$$\sum_m s_n(m-i)s_n(m)=\sum_{k=1}^{p}\hat{a}_k\sum_m s_n(m-i)s_n(m-k) \tag{3.14}$$

A covariância de curto prazo $\Phi_n(i,k)$ de s_n é dada por:

$$\phi_n(i,k)=\sum_m s_n(m-i)s_n(m-k) \tag{3.15}$$

A forma compacta de (3.15) é:

$$\phi_n(i,0)=\sum_{k=1}^{p}\hat{a}_k\phi_n(i,k) \tag{3.16}$$

Consequentemente, o mínimo erro quadrático pode ser expresso por:

$$\hat{E}_n=\sum_m {s_n}^2(m)-\sum_{k=1}^{p}\hat{a}_k\sum_m s_n(m)s_n(m-k)$$

$$=\phi_n(0,0)-\sum_{k=1}^{p}\hat{a}_k\phi_n(0,k) \tag{3.17}$$

Diferentes métodos podem ser utilizados para solucionar as equações $\Phi_n(i,k)$, expressas em (3.17).

Os principais são o método da autocorrelação e o método da covariância, apresentados a seguir.

No método da autocorrelação, uma forma eficaz de definir os limites de **m** é multiplicar o segmento de fala $s_n(m)$ por uma janela w(m) no intervalo descrito em (3.18):

$$s_n(m) = \begin{cases} s(m+n).w(m), 0 \le m \le N-1 \\ 0 \end{cases} \tag{3.18}$$

Desta forma, o erro passa a ser:

$$E_n = \sum_{m=0}^{N-1+p} e_n{}^2(m) \tag{3.19}$$

E a covariância passa a ser:

$$\phi_n(i,k) = \sum_{m=0}^{N-1+(1-k)} s_n(m) s_n(m+1-k), \quad \begin{matrix} 1 \le i \le p \\ 0 \le k \le p \end{matrix} \tag{3.20}$$

Por fim, a função de autocorrelação é definida por:

$$\phi_n(i,k) = r_n(i-k) = \sum_{m=0}^{N-1+(1-k)} s_n(m) s_n(m+1-k) \tag{3.21}$$

Considerando a simetria da autocorrelação, as equações do LPC podem ser descritas da seguinte forma:

$$\sum_{k=1}^{p} r_n(|i-k|)\hat{a}_n = r_n(i), \quad 1 \le i \le p \tag{3.22}$$

O método da covariância calcula o mínimo erro quadrático durante um intervalo **m,** sem a aplicação de pesos ao segmento de fala,

$$E_n = \sum_{m=0}^{N-1} e_n{}^2(m) \tag{3.23}$$

A função da covariância passa a ser:

$$\phi_n(i,k) = \sum_{m=-i}^{N-i-1} s_n(m)s_n(m+i-k), \quad \begin{matrix} 1 \le i \le p \\ 0 \le k \le p \end{matrix} \tag{3.24}$$

3.3.4.2. Energia de Curto Prazo

A energia de curto prazo Ec representa as variações de amplitude do sinal ao longo de tempo. Os intervalos vozeadas do sinal de fala apresentam maiores valores de energia de curto prazo que os intervalos não vozeados. É expressa por:

$$Ec(n) = \sum_{m=-\infty}^{\infty} x^2(m)\delta(n-m) \tag{3.25}$$

onde:

x(n): sinal de fala;

δ(n-m): impulso unitário.

3.3.4.3. Taxa de Cruzamento de Zero

A taxa de cruzamento de zeros Z(n) é definida por:

$$Z(n) = \sum_{m=-\infty}^{\infty} \left|\text{sgn}[x(m)] - \text{sgn}[x(m-1)]\right| w(n-m) \tag{3.26}$$

onde:

sgn[x(n)] = 1 se x(n) ≥ 0;

sgn[x(n)] = -1 se x(n) < 0;

s(n): amostra do sinal;

w(n) = 1/ 2n para 0 ≤ n ≤ (N-1) e w(n) = 0 fora do intervalo

A taxa de cruzamento de zero apresenta comportamento oposto ao da energia de curto prazo, pois apresenta altos valores em regiões não vozeadas do sinal de fala.

A taxa de cruzamento do zero e a energia de curto prazo são utilizadas em conjunto para delimitar os pontos iniciais e finais em uma locução, discriminando-as das zonas de silêncio.

Esse algoritmo de delimitação da região de sinal de fala, introduzido por Rabiner e Sambur (apud RABINER; SCHAFER, 1978), utiliza métodos estatísticos para definir limiares de taxa de cruzamento de zeros e energia de curto prazo. Os pontos de início e de fim da locução são definidos nos pontos em que esses limiares se cruzam ao longo da locução. (RABINER; SCHAFER, 1978).

3.3.4.4. MFCC (Mel-frequency Cepstral Coefficents)

Os coeficientes "*mel* - *cepstral*" são as características distintivas mais empregadas atualmente em reconhecimento de fala, principalmente em sistemas robustos ao ruído ambiente. São *features* baseadas em banco de dados. O entendimento dos MFCC pressupõe o conhecimento da escala mel, dos sistemas homomórficos e do cepstrum.

- escala de frequência *mel*

A escala *mel* é um mapeamento das freqüências fundamentais dos tons de áudio para um novo domínio, com base na sua percepção auditória, realizada por ouvintes juízes. O ponto de referência da maioria das versões dessa escala é (1.000 Hz, 1.000 mel), com um nível de 40 dB em relação ao limiar de percepção.

A escala foi proposta por Stevens, Volkman e Newman em 1937 (O'SHAUGHNESSY, 1987). A palavra "mel" é derivada de "melody".

Uma das mais conhecidas relações entre Hz e *mel*, mostrada na Fig.3.6, é expressa por:

$$m = 2595 \log [1 + (f / 700)] \qquad (3.27)$$

onde:

m: valor em mels;

f: frequência em Hz.

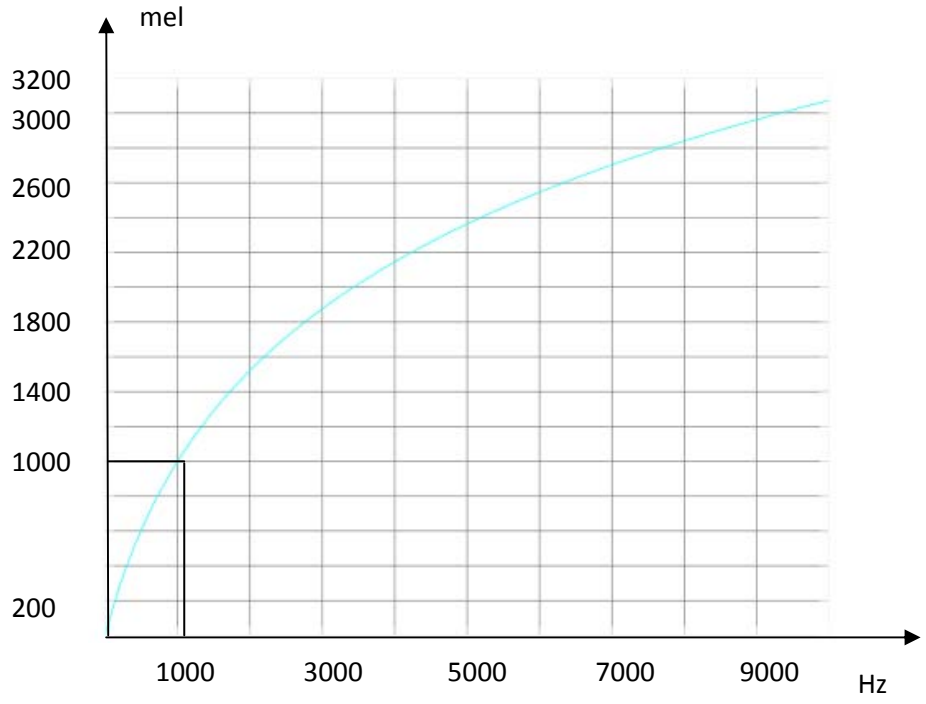


Fig.3.6. Escala de Frequência Mel

- sistemas homomórficos e *cepstrum*

São sistemas não lineares que atendem ao "princípio da superposição (e da escala) generalizada" (OPPENHEIN; SCHAFER, 1989), podendo ser representados por transformações lineares entre entrada e saída, expressas a seguir:

$$H\,[x_1(n)\ \mathbf{Si}\ x_2(n)] = H\,[x_1(n)]\ \ \mathbf{So}\ H[x_2(n)] \tag{3.28a}$$

$$H[k\ \mathbf{Mi}\ x(n)\,] = k\ \mathbf{Mo}\ x(n) \tag{3.28b}$$

onde:

Si: regra de combinação de entradas;

So: regra de combinação de saídas;

Mi: regra de combinação de entradas e escalar;

Mo: regra de combinação de saídas e escalar;

k: constante.

Um exemplo em que **Si** é a multiplicação e **Mi** é a exponenciação é o da função logaritmo:

$$\log\,[x_1(n)^{\,k1}\,.\ \ x_2(n)^{\,k2}] = \ k_1\,\log\,[x_1(n)]\ + k_2\,\log[x_2(n)] \tag{3.29}$$

onde:

ponto "." representa a multiplicação.

Em (3.29), visto que as entradas são complexas, ou seja, $x(n) = |x(n)|\ e^{j\ \arg[x(n)]}$, a função logaritmo deve ser também complexa. Considerando que o arg[x(n)] é uma função contínua de x, a função logaritmo complexa é assim definida:

$$\log[x(n)] = \log\,|x(n)| + j\,\arg\,[x(n)] \tag{3.40}$$

O *cepstrum* c(n) de um sinal de voz s(n) é obtido por uma transformação homomórfica cujo diagrama de blocos está mostrado na Fig.3.7.

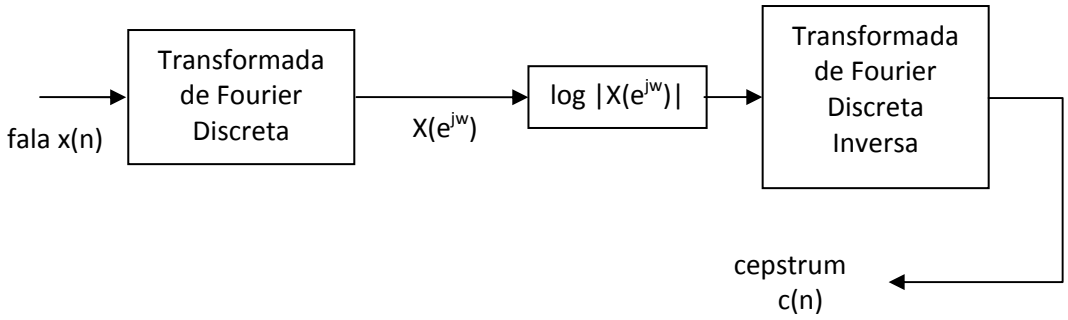


Fig.3.7. Obtenção do Cepstrum

- coeficientes "mel-cepstral"

Os coeficientes MFCC são obtidos através das seguintes fases (SKOWRONSKI, 2004):

1ª) O sinal de fala x(t) passa por um banco de filtros triangulares espaçados em escala mel.

As freqüências centrais dos filtros triangulares tentam imitar a característica do sistema auditório humano.

A Fig.3.8 ilustra um banco de filtros padrão, em que há uma superposição de 50% entre dois filtros adjacentes. Variações desse banco padrão têm produzido melhores resultados.

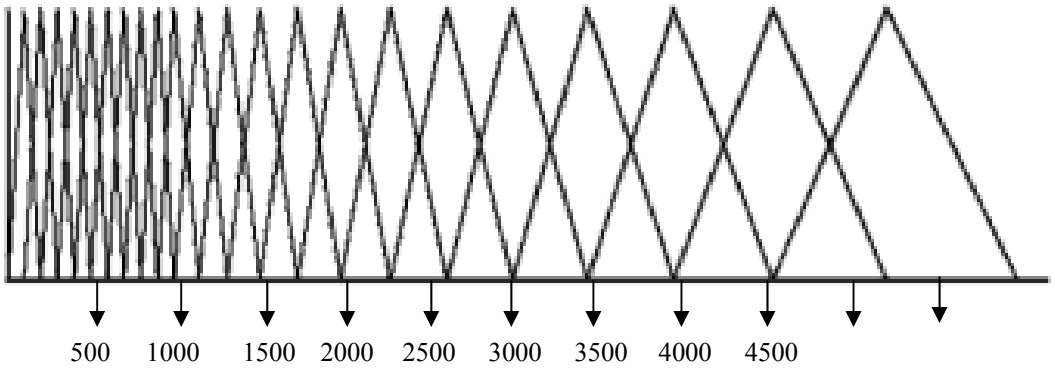


Fig.3.8 Banco de Filtros em Escala Mel

2ª) As energias do sinal proveniente da saída do banco de filtros são comprimidas pela aplicação do logaritmo.

3ª) Com a aplicação da Transformada Cosseno Discreto (DCT, *Discrete Cosine Transform*) são obtidos os coeficientes mel-cepstral. O espectro dos vetores de base do DCT são semelhantes ao espectro dos autovetores do sinal obtido na 2ª fase (diferença da média do espectro menor que 15%).

Os MFCC são expressos por:

$$\text{MFCC i} = \sum_{k=1}^{N} X_k \cos[i(k - \frac{1}{2})\frac{\pi}{N}] \tag{3.41}$$

onde:

X_k: logaritmo da saída de energia do k-ésimo filtro;

N: número de filtros;

i: 1, 2, ..., M, sendo M o número de coeficientes "mel-*cepstral*" ;

CAPÍTULO 4 - PROJETO E DESENVOLVIMENTO DO RECONHECEDOR DE COMANDOS

4.1. Introdução

As fases do projeto e desenvolvimento do reconhecedor de comandos estão mostradas na Fig.4.1. Os comandos de voz do vocabulário foram: Direita, Esquerda, Frente, Trás, Pare e Desligue.

As amostras foram coletadas pelo critério de conveniência para permitir a generalização relativa a sexo e idade dos locutores.

O pré-processamento consistiu na normalização das amostras, filtragem do sinal de fala, determinação dos pontos extremos e segmentação.

As *features* testadas e utilizadas foram a taxa de cruzamento do zero, a energia de curto prazo, os coeficientes LPC e o seu erro e os coeficientes mel-ceptrum.

O classificador foi uma rede neural *perceptron* multicamadas, treinada com o algoritmo *backpropagation* e alimentada em batelada (*batch*).

A verificação da taxa de acerto foi realizada por simples correlação cruzada entre os resultados obtidos e os esperados.

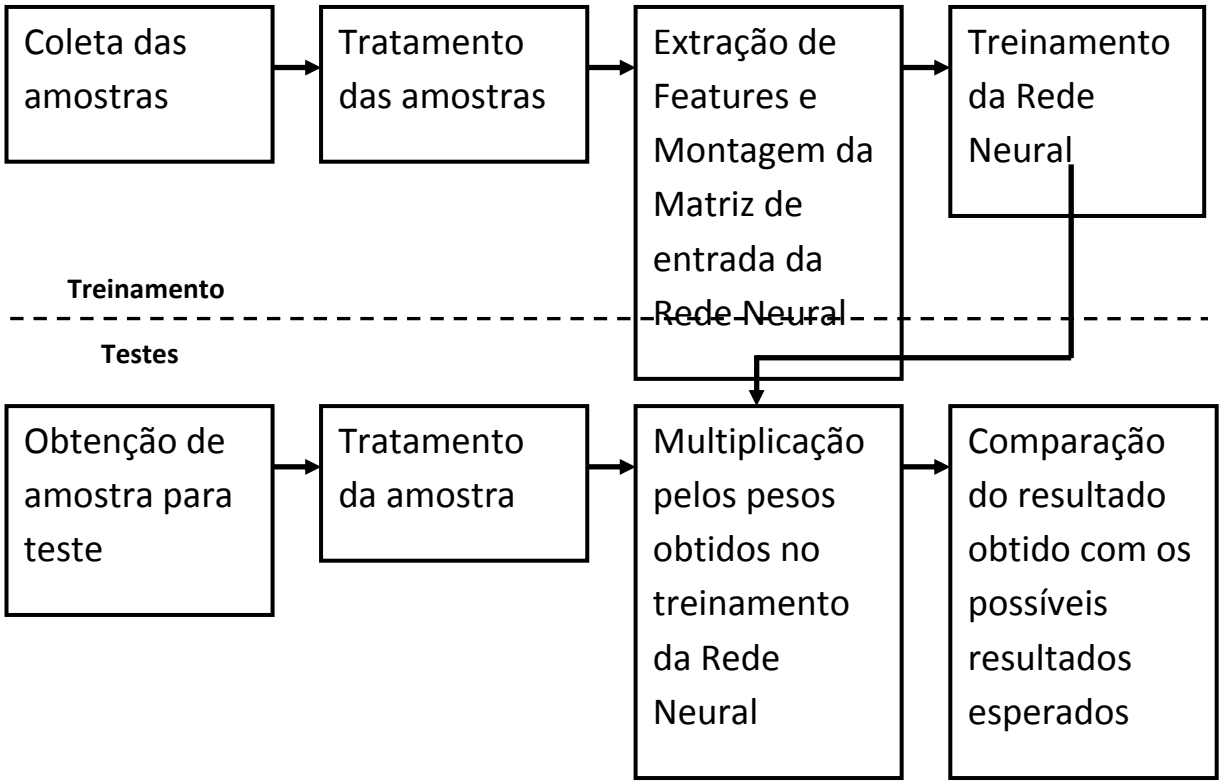


Fig. 4.1. Fases do Desenvolvimento do Reconhecedor

4.2. Coleta das Amostras dos Comandos

A coleta das amostras dos comandos empregou um grupo experimental e outro de controle, de conformidade com as exigências do treinamento supervisionado.

O caráter de conveniência da amostra em relação à diversificação da idade e sexo dos locutores teve por objetivo garantir uma maior discrepância entre as características de voz dos participantes, buscando alcançar a generalização da rede neural utilizada no sistema. Além disso, as amostras dos comandos de voz foram coletadas em diversas situações, durante o dia ou noite, em diversos ambientes e circunstâncias.

A Tabela 4.1 resume os dados referentes à amostra.

O total de amostras de comandos de voz desta pesquisa foi de 80, 60 pertencentes ao grupo experimental (grupo de treinamento da rede neural), e 20 ao grupo de controle (grupo de teste da rede neural). Esse número de amostras é compatível com outras pesquisas dessa natureza (THIANG; WIJOYO, 2011).

O grupo *experimental* contou com 7 representantes do sexo masculino e 5 do sexo feminino. O grupo *controle* foi composto por 3 homens e 1 mulher.

A distribuição por faixa etária foi de 10 locutores de 12 a 20 anos, 15 de 21 a 24 anos e 55 de 41 a 60 anos.

Tabela 4.1 Distribuição das amostras (comandos) por sexo e idade

Grupo	Sexo	12 a 20 anos	21 a 40 anos	41 a 60 anos	totais
controle	masculino	5	0	10	15
(teste)	feminino	0	0	5	5
experimental	masculino	5	5	25	35
(treinamento)	feminino	0	10	15	25
totais	-	10	15	55	80

Os instrumentos eletrônicos e aplicativos utilizados para o desenvolvimento do sistema foram:

- Laptop Dell Vostro e um Laptop Dell Latitude, ambos com sistema multimídia de fábrica, com microfones embutidos: utilizados na coleta de dados e em todas as outras fases da pesquisa;

- Aplicativo Audacity®: para a digitalização do sinal de voz;

- Aplicativo Matlab®: para o desenvolvimento dos programas de pré-processamento, extração de *features* e de classificação;

- Microcontrolador PIC 16F: para controle da fase de visualização dos resultados através de LED´s.

A pesquisa foi efetuada no Laboratório de Processamento de Sinais do Departamento de Engenharia Elétrica da Universidade de Taubaté (UNITAU).

O software Audacity® foi escolhido para a realização das coletas, pois possui todos os recursos necessários à tarefa.

As locuções coletadas foram armazenadas em arquivos do tipo wave (.wav) com taxa de amostragem de 44,100 kHz, 16 bits, modulação PCM (*Pulse Code Modulation*) 16 bit e foram devidamente testadas quanto à sua integridade. As amostras foram armazenadas em pastas de acordo com o seu grupo (experimental ou controle) e o comando que representam.

4.3. Pré-processamento (tratamento das amostras de voz)

- normalização

Cada amostra de um comando de voz é carregada em um vetor de amplitudes, normalizado para valor máximo igual a 1. As amostras de fala são retiradas dos seguintes comandos de voz: *direita, esquerda, frente, trás, pare e desligue*.

- determinação dos pontos extremos

A seguir é realizada a determinação dos pontos extremos da locução (comando), para evitar que os trechos onde há apenas ruído sejam processados (Fig.4.2).

Os limiares dos pontos extremos são estimados pela energia de curto prazo (item 3.3.4.2) e pela taxa de cruzamento de zero (item 3.34.2), por meio de procedimentos experimentais. Orienta o algoritmo o fato de a energia de curto prazo ser baixa e a taxa de cruzamento de zero ser alta para o ruído (ou sinais não vozeados).

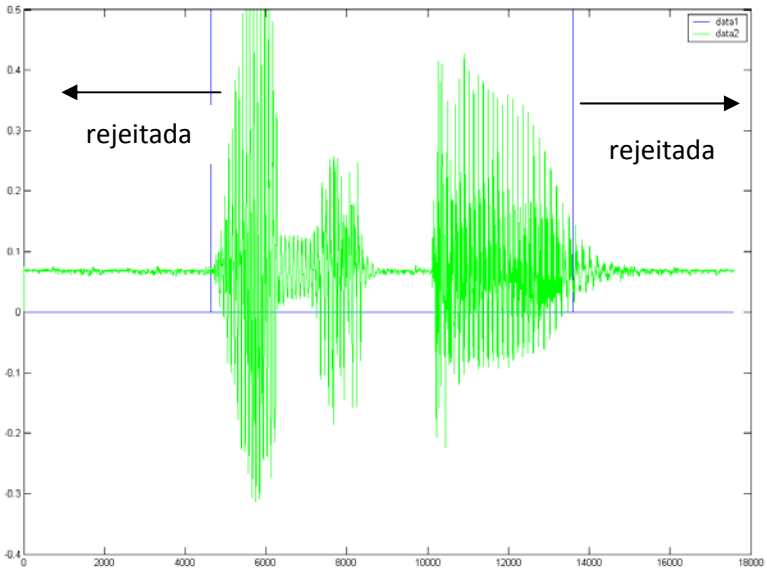


Fig.4.2. Marcação dos Pontos Extremos em uma Amostra de Fala

Um primeiro procedimento consiste no seguinte:

- Obtém-se a soma da energia de curto prazo Ec_T e a soma das taxas de cruzamento de zero Z_T dos quatro primeiros segmentos (quadro) do sinal de comando, que representam o nível de silêncio;
- Estabelece-se o limiar de energia de curto prazo como $Ec_{limiar} = Ec_T \times 0,5$ e o limiar da taxa de cruzamento de zero como $Z_{limiar} = 2\ Z_T$;
- Se em um quadro de ordem k, $Z_k \geq Z_{limiar}$ e $Ec_k \leq Ec_{limiar,}$ então o quadro será considerado de silêncio (ruído).

Outros procedimentos utilizam os primeiros 100 ms de locução (comando) para estimar a média e o desvio padrão e verificar se um valor prático mínimo de Z_{limiar}, está no intervalo de confiança da variável aleatória "média estimada". O limiar para energia de curto prazo é tomado como quatro vezes o valor obtido nos primeiros 200 ms da locução.

Neste trabalho foram utilizados os 200 ms iniciais tanto para o limiar da energia de curto prazo quanto para o da taxa de cruzamento de zero. O valor do limiar da taxa de cruzamentos de zero foi $Z_{limiar} = 25$. Para a energia de curto prazo o limiar Ec_{limiar} utilizado foi:

$$Ec_{limiar} = (Emax - Emin) /10 \qquad (4.1)$$

onde:

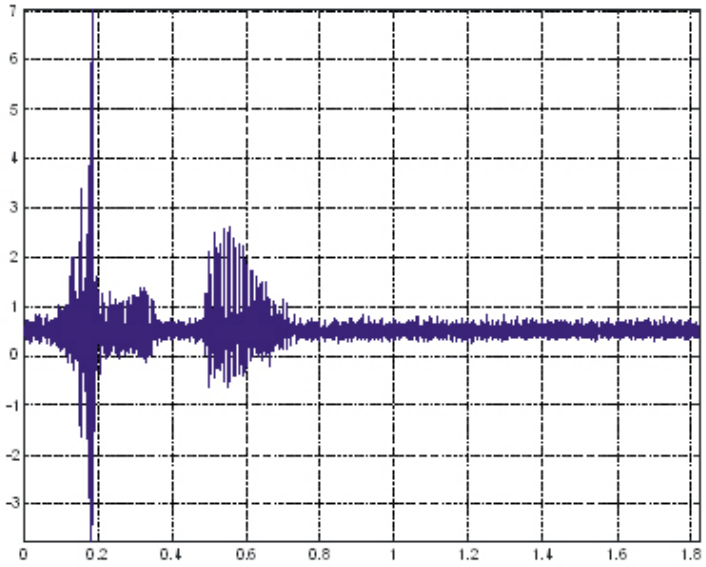
Emax: máxima energia nos 200 ms iniciais da locução;

Emin: mínima energia nos 200 ms iniciais da locução

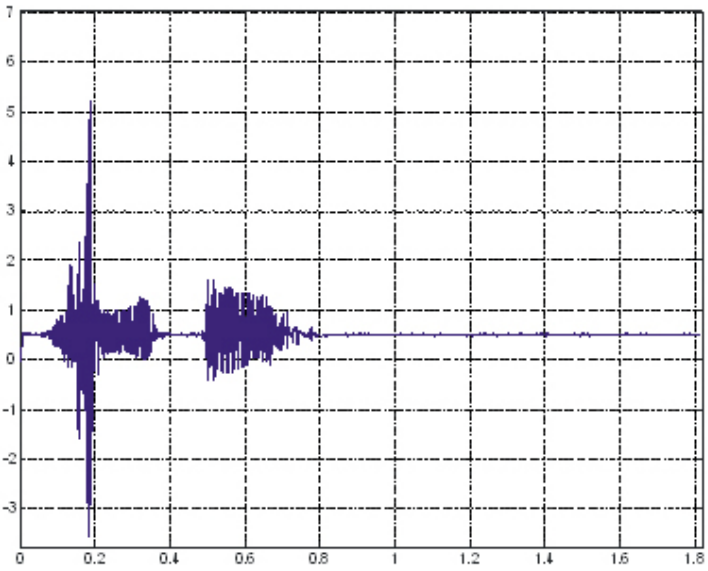
- filtragem

A filtragem do sinal de fala antes da extração das *features* possibilita também a eliminação do ruído externo.

O experimento inicial consistiu na utilização de um filtro FIR (*Finite Impulse Response*), passa baixa com frequência de corte em 3 kHz (levando em consideração que 4kHz é a largura de faixa efetiva do sinal de voz) (McLOUGHLIN, 2009) e 97 coeficientes, cujo resultado está ilustrado nas Fig.4.3 e Fig.4.4.

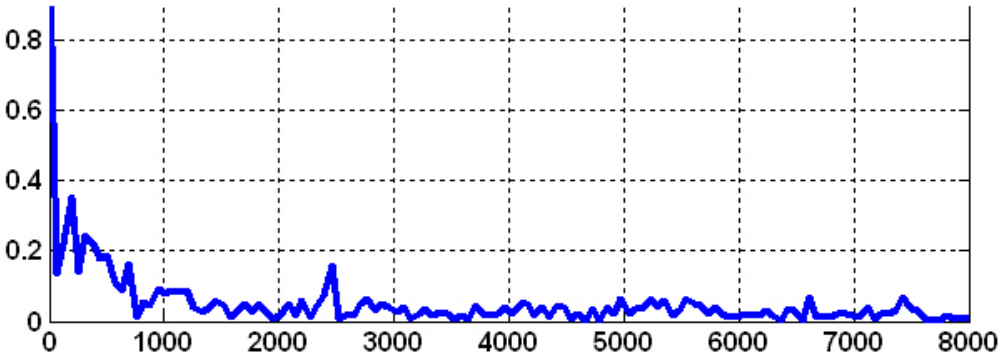


(a) Sinal de Voz no Domínio do Tempo Antes da Filtragem.

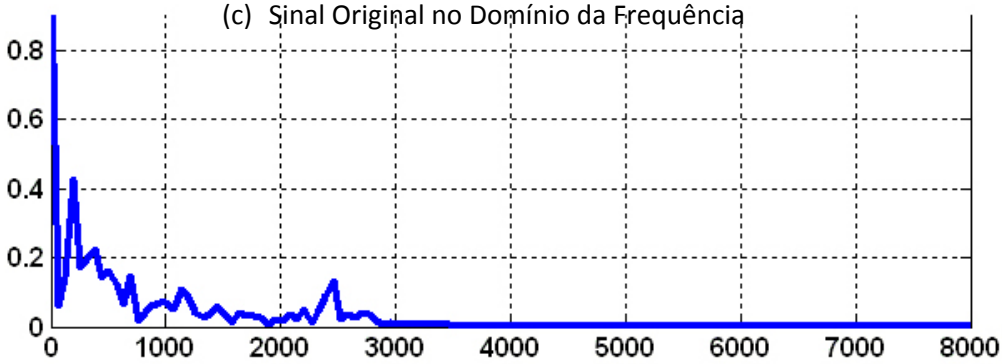


(b) Sinal de Voz no Domínio do Tempo Depois da Filtragem

Fig. 4.3. Resultado da Aplicação do Filtro FIR no Domínio do Tempo



(c) Sinal Original no Domínio da Frequência



(d) Sinal Filtrado no Domínio da Frequência

Fig.4.4. Resultado da Aplicação do Filtro FIR no Domínio da Frequência

A alternativa que produziu melhores resultados para a posterior tarefa de extração das *features* foi a utilização de um filtro adaptativo de Wiener, com vizinhança de 10 amostras.

4.4. Segmentação do Sinal e Janelamento

O sinal de comando de fala saída do filtro foi dividido em 200 segmentos de igual duração, por meio da janela de Hamming, com uma superposição de 25% entre janelas adjacentes.

A janela de Hamming, mostrada na Fig.4.5, é definida por:

$$w(n) = 0{,}54 - 0{,}46\cos(2\pi n / N) \quad \text{para} \quad 0 < n \leq N-1$$

$$= 0 \quad \text{para} \quad n \leq 0 \ \text{ e } n > N-1 \qquad (4.2)$$

onde:

w(n): janela de Hamming;

N: número total de pontos.

Uma vez que os comandos têm durações diferentes e o número de segmentos é fixo (para formar o mesmo número de entradas na rede neural), as durações das janelas de cada sinal de comando de fala são diferentes.

A utilização da janela de Hamming com as *features* "coeficientes e erro LPC" foram semelhantes aos obtidos com a janela retangular. Já para as *features* "mel-cepstral" (item 3.3.4.4) a janela de Hamming revelou-se mais adequada.

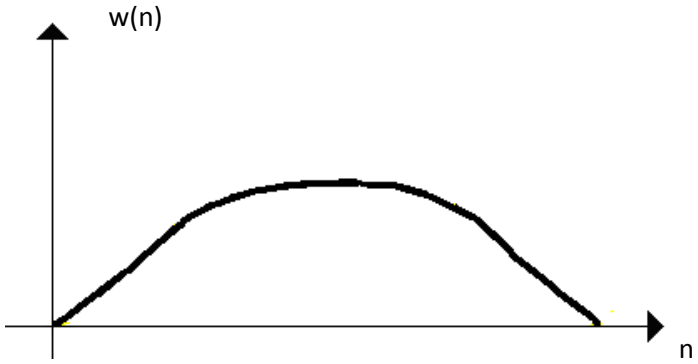


Fig.4.5 Janela de Hamming

4.5. Extração das Features

- primeiro experimento

As *features* utilizadas na primeira experimentação foram a energia de curto prazo (item 3.3.4.2), taxa de cruzamento de zeros (item 3.3.4.3), os dois primeiros coeficientes LPC e o erro do LPC (item 3.3.4.1).

A utilização de mais coeficientes LPC não proporcionou nenhuma melhora de desempenho do reconhecedor.

As *features* extraídas de cada janela foram armazenadas em vetores, os quais foram organizados de forma a construir a matriz de entrada em batelada para o treinamento da rede neural, como mostra a Fig.4.6.

O erro relativo percentual total foi de 13,33% e a taxa de acertos foi de 86,66%, conforme Tabela 5.1. (descrita em Resultados).

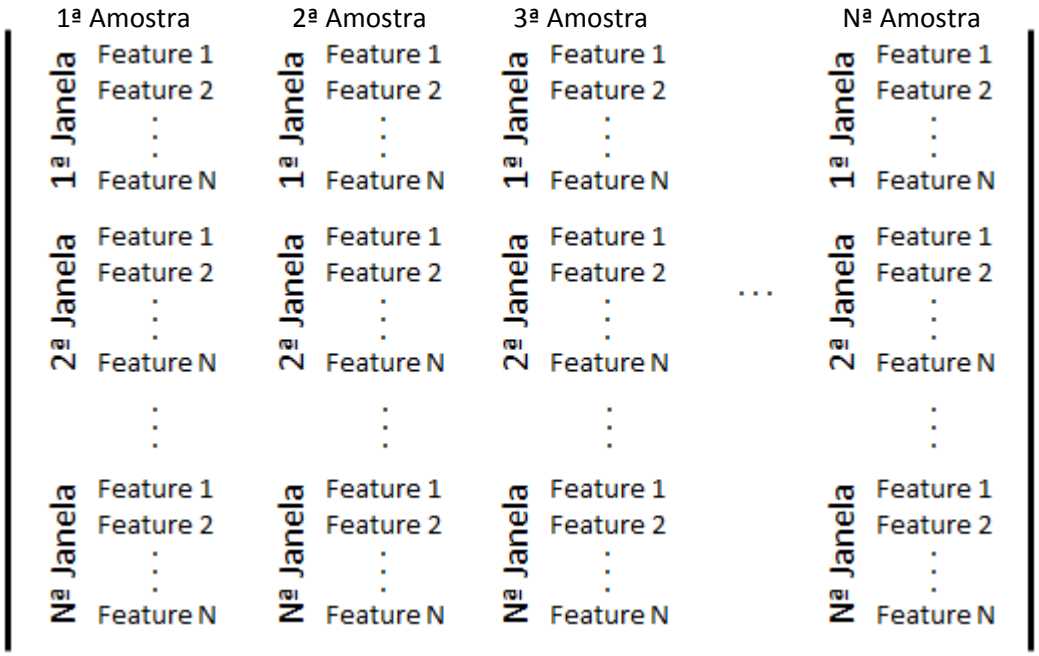


Fig.4.6. Matriz de Entrada em Batelada para treinamento da Rede Neural no Primeiro Experimento.

- segundo experimento

Nesse segundo experimento foram utilizados os coeficientes “mel-cepstral” (MFCC , *Mel - frequency Cepstral Coefficients*).

A matriz desse experimento é a mesma da Fig.4.6., substituindo os três coeficientes LPC pelos coeficientes “mel-cepstral”.

O erro relativo percentual caiu a 10,83 e a taxa de acertos aumentou para 89,16%, conforme Tabela 5.2. (descrita em Resultados).

4.6. Treinamento da Rede

A estrutura da rede neural escolhida para efetuar a tarefa do reconhecimento foi um *perceptron* multicamadas, treinado com o algoritmo *backpropagation*.

Foram experimentados diferentes números de camadas escondidas e diferentes números de neurônios por camada. O melhor resultado foi um *perceptron* de duas camadas escondidas, com 32 neurônios por camada.

Em uma primeira experimentação utilizou-se a função de ativação tangente hiperbólica para as camadas escondidas. Na camada de saída foi utilizada a função linear saturada em -1 e 1 e o treinamento foi efetuado até atingir o número máximo de iterações 50.000 ou o erro mínimo de 0,0001. A taxa de aprendizagem foi 0,1.

Em uma segunda experimentação foram utilizados os mesmos parâmetros e a mesma estrutura, porém a função linear saturada, na camada de saída, foi substituída por uma função tangente hiperbólica, ficando, dessa forma, as três camadas com a mesma função de ativação.

Foram definidos códigos de 8 bits para os comandos. Cada código constitui um vetor coluna da matriz de alvos $A_{8 \times 6}$ (valores de saída desejados), mostrada na Fig.4.7.

A matriz de saída da rede neural S_{8x6} é formada por 6 colunas de 8 bits, correspondendo cada coluna àquela de mesma ordem da matriz de entrada (cada coluna para um comando). Assim, na condição ideal de função de erro nula, a matriz S_{8x6} seria idêntica à matriz de alvos $A_{8 \times 6}$.

$A_{8 \times 6} =$

Direita	Esquerda	Frente	Tras	Pare	Desligue
1	1	1	1	1	-1
1	1	1	1	-1	1
1	1	-1	-1	1	-1
1	1	-1	-1	-1	1
1	-1	1	-1	1	-1
1	-1	1	-1	-1	1
1	-1	-1	1	1	-1
1	-1	-1	1	-1	1

Fig.4.7. Matriz de Códigos dos Comandos (Alvos)

Ao trabalhar com redes neurais para reconhecimento de padrão, obtém-se melhor resultado do treinamento se os dados de mesmo tipo não forem apresentados à rede consecutivamente (HAYKIN, 1999).

A rede neural foi construída com o Neural Network Toolbox do aplicativo Matlab®, utilizando os comandos *newff* e *train*.

4.7. Teste da Rede

Durante a etapa de teste, as amostras de comando do grupo controle (de teste) passam pelo mesmo tratamento (pré-processamento, processamento e extração de *features*) que as amostras de comando do grupo de treinamento experimentaram. Um específico comando de ordem **j** (j = 1, 2, 3, 4, 5, 6) gera um vetor coluna **j** de *features* do mesmo tamanho das colunas da matriz de entrada da rede neural.

Com a entrada do vetor coluna de ordem j de *features* na rede neural já treinada, ela fornece, na saída, um vetor coluna de 8 bits S_j.

É realizada então a correlação cruzada C_j entre esse vetor S_j e todos os seis códigos alvos definidos para os comandos (colunas A_j da matriz da Fig.4.8).

Conforme evidencia a expressão (4.3) da correlação cruzada C_j, o comando recebido de ordem **j** é identificado como aquele que tem o maior valor C_j.

$$C_j = \sum_{i=1}^{8} S_{i,j} A_{i,j} \tag{4.3}$$

4.8. Atuação do Comando

Foi construído um circuito simples, ilustrado na Fig.4.8., para emular a atuação dos comandos em uma máquina, por exemplo, um robô.

Cada comando é representado por um caractere, conforme a Tabela 4.2.

Tabela 4.2 Caracteres e Portas para cada Comando de Voz.

Comando	Caractere	Número da Porta	Porta
Direita	A	40	RB7
Esquerda	B	39	RB6

Frente	C	38	RB5
Trás	D	37	RB4
Pare	E	36	RB3
Desligue	F	35	RB2

Ao identificar um comando, o caractere correspondente é enviado à porta serial do microcomputador, para acionar um display de LED´s (ALVARENGA, 2010), sob o controle do microcontrolador PIC16F877 (MICROCHIP).

A interface do microcontrolador com a porta serial RS 232 é realizada pelo circuito integrado MAX232 (Dual EIA-232 *driver /receiver* da MAXIM - *Dallas Semiconductor*).

O microcontrolador PIC16F877 foi programado em linguagem C, através do compilador CCS C (*Custom Computer Services* – C).

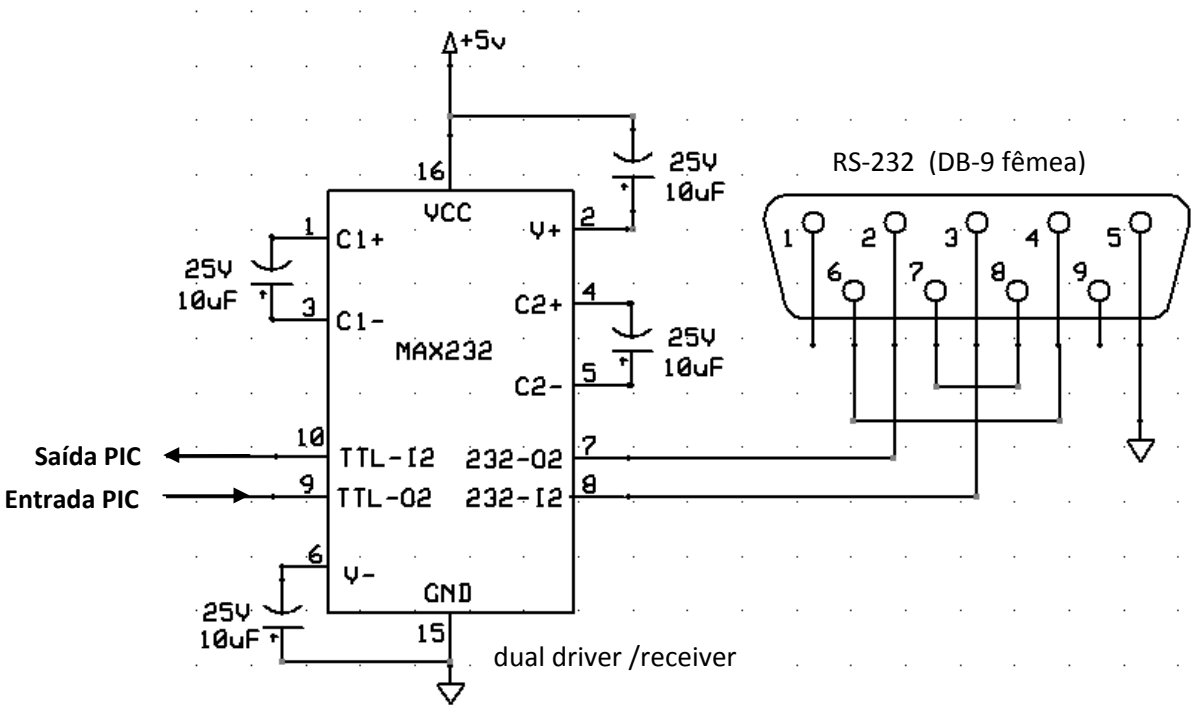


Fig.4.8. Circuito de Emulação do Reconhecedor

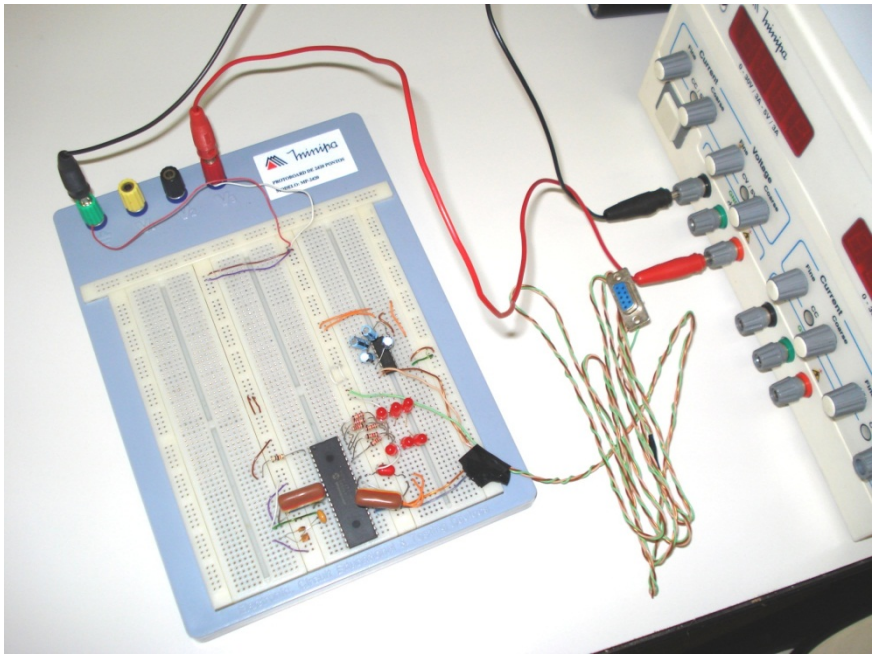


Fig.4.9. Montagem do Circuito de Emulação do Reconhecedor

CAPÍTULO 5 - RESULTADOS E CONCLUSÕES

5.1 Resultados

Os resultados parciais serão apresentados nas diversas fases do reconhecedor.

- fase de coleta de amostras (comandos)

Para o treinamento foram coletadas amostras de comandos proferidos por locutores pertencentes a três faixas de idade, e do sexo masculino e feminino. O grupo de teste não está contido no grupo de treinamento. As condições foram diversificadas visto que as amostras dos comandos foram coletadas em diversas situações e condições ambientais. O objetivo foi a obtenção da generalização da rede neural. Trata-se do caso mais difícil de reconhecimento (projeto de "pior caso" da taxa de acertos), completamente independente do locutor do mesmo idioma.

Em sistemas dependentes de locutores pertencentes a um pequeno grupo, por exemplo, conjunto das pessoas autorizadas a comandar uma máquina ou robô, a taxa de acertos pode chegar a 100%. Neste caso, o grupo de treinamento e o grupo de teste são os mesmos.

- fase de pré-processamento

A normalização é imprescindível.

A determinação dos pontos extremos guiou-se pelos valores da energia de curto prazo e da taxa de cruzamento de zero. A dificuldade do método aumenta quando a locução começa por um fonema não vozeado fricativo, como o /f/ do comando "frente", em que o sinal pode confundir-se com o ruído.

Os limiares foram ajustados experimentalmente conforme o item 4.2., fornecendo bons resultados.

A filtragem ajuda na delimitação dos pontos extremos do sinal de fala e confere maior robustez ao reconhecimento do comando na presença de ruído ambiental. O filtro adaptativo

Wiener mostrou-se mais eficiente que o filtro FIR por eliminar a maior parte do ruído sem comprometer a inteligibilidade das amostras dos comandos.

Para o experimento com as *features* "mel-cepstral" a janela de Hamming foi a mais adequada, para a superposição dos segmentos adotada (25%).

- fase de extração das *features*

É a etapa mais sensível do reconhecimento de fala. A primeira experimentação utilizou os seguintes parâmetros: energia de curto prazo, taxa de cruzamento de zeros, dois primeiros coeficientes LPC e o seu erro. A experimentação de acrescentar mais parâmetros, tais como, o terceiro coeficiente LPC e a duração da locução, não redundou em melhoria significativa da taxa de acertos. Os resultados foram razoáveis, com a obtenção de um erro relativo percentual total de 13,33%, e uma percentagem de acerto de 86,66%, conforme Tabela 5.1.

Tabela 5.1. Resultados com os Coeficientes LPC

Comando	Testes	Número de Acertos	Acertos (%)	Erro relativo percentual (%)
Direita	20	16	80,00	20,00
Esquerda	20	19	95,00	5,00
Frente	20	19	95,00	5,00
Trás	20	19	95,00	5,00
Pare	20	18	90,00	10,00
Desligue	20	13	65,00	35,00
Total	120	104	86,66	13,33

Na segunda experimentação, utilizando os coeficientes "mel-cepstral", o erro relativo percentual total caiu a 10,83 e a percentagem de acerto total subiu a 89,16%.

Tabela 5.1. Resultados com os Coeficiente Mel-cepstral

Comando	Testes	Número de Acertos	Acertos (%)	Erro relativo percentual (%)
Direita	20	14	70,00	8,57
Esquerda	20	19	95,00	5,00
Frente	20	20	100,00	0,00
Trás	20	18	90,00	10,00
Pare	20	19	95,00	5,00
Desligue	20	17	85,00	15,00
Total	120	107	89,16	10,83

- estrutura e treinamento da rede

Foi realizada uma intensa, porém, não exaustiva, experimentação relativa ao número de camadas escondidas, número de neurônios por camada e tipos de função de ativação. A melhor configuração obtida foi um *perceptron* de duas camadas escondidas com 32 neurônios por camada e com ativação de todas as camadas, incluindo a de saída, pela função tangente hiperbólica.

A Fig.5.1 mostra o resultado do treinamento dessa rede *perceptron* com ativação da camada de saída por uma função linear saturada em -1 e 1 e das camadas escondidas pela função tangente hiperbólica. A taxa de aprendizagem foi 0,1 e o número de iterações, 50.000. O erro obtido após a última iteração foi de 0,00625.

A Fig.5.2 mostra o resultado do treinamento dessa mesma rede *perceptron* com a mesma taxa de aprendizagem 0,1 e o mesmo número de iterações, mas com a ativação de todas as camadas pela função tangente hiperbólica. O erro obtido após a última iteração foi 0,00167.

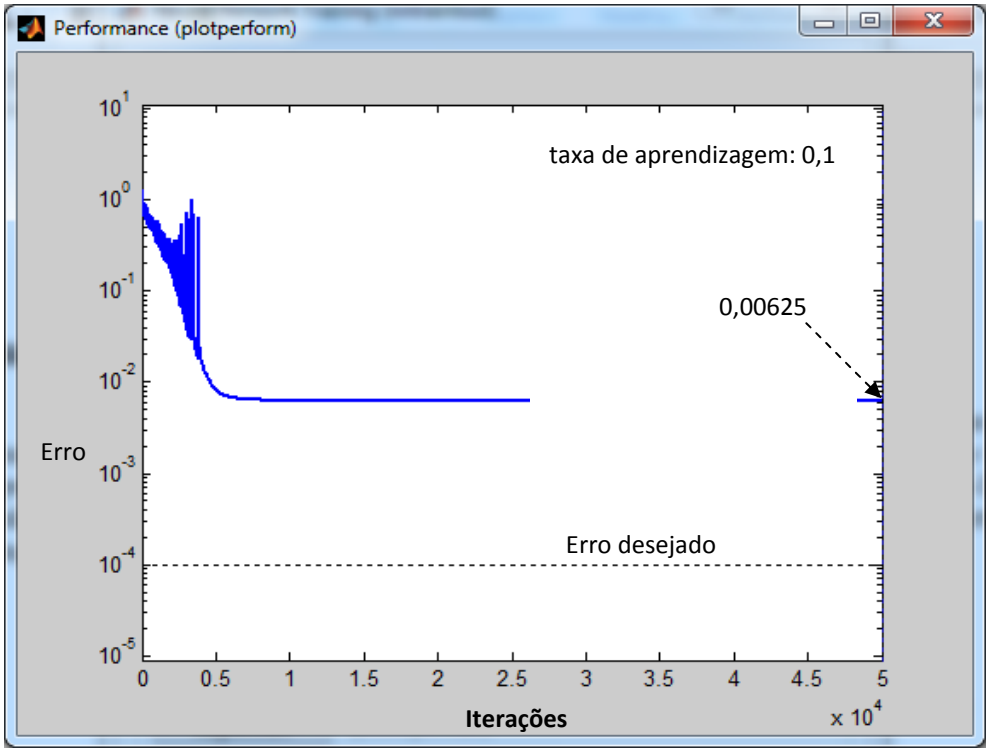


Fig.5.1. Treinamento com Ativação da Camada de Saída por Função Linear Saturada.

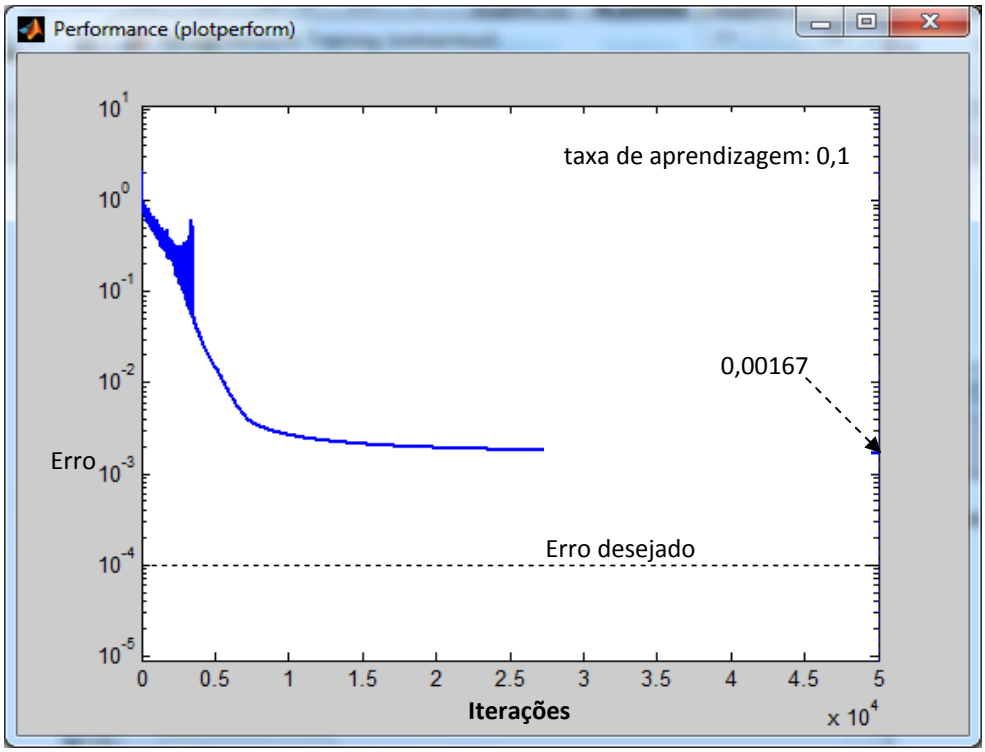


Fig.5.2 Treinamento com Ativação de todas as Camadas por Função Tangente Hiperbólica

- comparação da taxa de acertos

As referências atuais em reconhecimento de palavras isoladas registram taxas de aceitação (ou de reconhecimento) em torno de 90%. Thiang e Wijoyo (2011) divulgam uma taxa de acerto máxima de 91,4 % utilizando LPC e distância euclidiana, para o idioma indonésio. As condições de coleta das amostras e treinamento, no entanto, parecem, smj, menos adversas que a do presente trabalho.

5.2. Conclusões

Os resultados obtidos comprovam que o objetivo específico da dissertação foi atingido.

O sistema de reconhecimento de fala com base nas *features* "mel-cepstral" aproxima-se bem do estado da arte em reconhecimento de palavras, independentemente do locutor, tendo conseguido uma excelente taxa de acertos, na condição estabelecida de "pior caso" de amostragem dos comandos.

A pesquisa, no entanto, pode avançar na seguinte direção:

- experimentar novos classificadores baseados em redes neurais;

- testar novos valores das frequências centrais e principalmente das bandas dos filtros utilizados na obtenção dos coeficientes "mel-cepstrum";

- incorporar técnicas para aumentar a robustez dos classificadores ao ruído, tendo por base a modificação da fala das pessoas quando emitem uma locução em ambientes ruidosos.

REFERÊNCIAS

ANDERSON J. A.; et al. *The brain-state-in-a-box (BSB) neural model,* Psychological Review, 1977.

ALVARENGA, R.J. *Reconhecimento de palavras isoladas para comando de robô.* Trabalho de Conclusão de Curso (Graduação em Engenharia de Telecomunicações). Universidade de Taubaté, Taubaté, São Paulo, 2010.

BEALE, M. H.; HAGAN, M. T.; DEMUTH, H. B. *Neural Network Toolbox.* The MathWorks, Inc: 2010.

BEZERRA, M. R. *Reconhecimento automático de locutor para fins forenses utilizando redes neurais*. Dissertação de Mestrado (Mestrado em Engenharia Elétrica). Instituto Militar de Engenharia (IME). Rio de Janeiro, 1994.

BRAGA, A. P.; CARVALHO, A. P. L. F.; LUDEMIR, T. B. *Redes neurais artificiais: teoria e aplicações*, 2007.

CARRARA, V. *Redes Neurais Aplicadas ao Controle de Altitude de Satélites com Geometria Variável,* Tese de Doutorado, INPE, São José dos Campos, São Paulo, 1997.

CASTRO, A. A. M. *Algorítmos para Reconhecimento de Padrões.* Dissertação de mestrado Universidade de Taubaté, Taubaté, São Paulo, 2001

FALAUNASP. Disponível em: <http://falaunasp.wordpress.com/a-teoria/fonema/>. Acesso em 20 nov. 2011.

HAYKIN, S. *Redes neurais - princípios e prática*. 2 ed. Bookman, 1999.

HEBB, D.O. *The Organization of Behavior,* New York: Wiley, 1949.

HOPFIELD, J. J. Neural networks and physical systems with emergent colletive computational abillities. *Proceedings of National academy of Science*, USA, v. 79, 1982

KATAGIRI, S. *Handbook of neural networks for speech processing*. Londres: Artech House, 2000.

KOSKO, B. Adaptative Bidirectional Associative Memories. *Applied Optics*, v. 7, 1987

MATLAB®. *Conjunto de programas: version 6.00.88 release 12*. The MathWorks, Inc: Sep.22, 2000. Copyright 1984-2000. 1 CD.

McLOUGHLIN, I. *Applied speech audio processing*. Cambridge University Press Now Publishers, 2009.

McCULLOCH, W; PITTS, W. A Logical Calculus of the Ideas immanent in nervous activity. *Bulletin of Mathmatical and Biophysics*, 1943

MINSKY, M.; PAPERT, S. *Perceptrons: An Introduction to Computational Geometry.* Cambridge: MIT Press, 1969

OPPENHEIN, A. V. SCHAFER, W. S. *Digital signal processing*. Prentice Hall: Englewood Cliffs, New Jersey, 1989.

O'SHAUGHNESSY, D. *Speech communication: human and machine*. Addison-Wesley, 1987. ISBN 978-0-201-16520-3.

RABINER, L. R.; JUANG, B. H. *Fundamentals of speech recognition*. Prentice Hall: Englewood Cliffs, New Jersey, 1993.

RABINER, L.R.; SCHAFER, R.W. *Digital processing of speech signals*. Prentice Hall: Englewood Cliffs, New Jersey, 1978.

ROSENBLATT, F. *Principles of Neurodynamics.* Washington: Spartan books, 1962

SILVA, I. N.; SPATTI, H. D.; FLAUZINO, R. A. *Redes neurais artificiais para engenharia e ciências aplicadas,* São Paulo: Artliber, 2010.

SKOWRONSKI, M. D.; HARRIS, J.G. Increased MFCC bandwidth for noise-robust phoneme recognition. *ICASSP 02*, v.1, p. 801-4, 2002. ISBN 0-7803-7403-7.

THIANG; WIJOYO, S. *Speech recognition using linear predictive coding and artificial neural network for controlling movement of mobile robot,* Surabaya: IACSIT Press, 2011.

TOU, J. T.; GONZALEZ, R. C. *Pattern recognition principles*, Massachusetts: Addison Wesley , 1981.

WIDROW, B., HOFF, M. E. *Adaptive switching circuits,* 1960 IRE WESCON Convention Record, New York IRE, 1960