RECONHECEDOR DE VOZ VIA REDES NEURAIS

Daniel Richetti Lemos[1]
Gabriel Junqueira Rodrigues[2]
Emílio Del Moral Hernandez[3]

Resumo

Este projeto propõe uma solução para a tarefa de reconhecimento de voz utilizando redes neurais. A utilização do modelo de neurônios artificiais resulta em um sistema robusto com baixa taxa de erros de reconhecimento. A robustez do sistema é resultado da característica intrínseca de adaptabilidade das redes neurais, tanto maior o uso maior a robustez.

O projeto trabalha com um conjunto pré-definido de palavras e pode ser utilizado para a realização de comandos de voz em qualquer tipo de sistema, ou ainda, como cadeado eletrônico via voz.

Palavras Chave: Redes Neurais; Reconhecimento de Voz; Classificação de Padrões; Processamento de Sinais de Voz; Banco de Filtros.

Abstract

This project propose a solution for the voice recognition task, utilizing neural networks. The use of artificial neuron model results on a robust system with low miss-recognition rates. The system robustness comes from the intrinsic adaptability characteristic of the neural networks, the more the use the more the robustness.

The project works with a pre-defined set of words and it can be used to give voice commands to any kind of system and also as an electronic lock.

Key words: Neural Networks; Voice Recognition; Pattern Recognition; Voice Signal Processing; Filter Bank.

1. Introdução

A proposta do projeto foi realizar o reconhecimento de voz utilizando redes neurais. Pode-se fazer uma analogia entre o sistema desenvolvido e o mecanismo de interpretação de sons do corpo humano. Assim o projeto pode ser dividido em duas partes: o cérebro e o ouvido. O ouvido é responsável por captar sons e transformá-los de maneira adequada a facilitar o reconhecimento. O cérebro é o órgão que efetivamente

[1] Aluno de graduação de Departamento de Sistemas Eletrônicos da EPUSP. Trabalho de Iniciação Científica sem bolsa.
[2] Aluno de graduação de Departamento de Sistemas Eletrônicos da EPUSP.
[3] Professor do Departamento de Sistemas Eletrônicos da EPUSP.

realiza o reconhecimento, aprendendo ao longo da vida como distinguir padrões diferentes.

O reconhecimento de voz não é uma tarefa que pode ser facilmente implementada computacionalmente. Há um número muito grande de parâmetros envolvidos em um sinal de voz, por exemplo: timbre, cadência, duração, volume além de variações intencionais da pronúncia. A mesma palavra nunca é pronunciada exatamente igual duas vezes. A variação existente entre duas amostras é ainda maior entre locutores distintos. Então, quando os sinais de voz devem ser considerados iguais? O mecanismo utilizado pelo cérebro para transformar os impulsos neurais provocados pelo som em informações lingüísticas não é tão claro. Não se sabe de antemão quais são as variações de características aceitas, de modo que a palavra continue a ser igualmente interpretada. A dificuldade de se realizar computacionalmente o reconhecimento reside no fato de ser impossível determinar claramente os critérios de distinção de palavras.

As redes neurais são modelos computacionais que simulam o comportamento dos neurônios biológicos. Podem realizar diversos tipos de funções, e têm muito bom desempenho na resolução de problemas de classificação de padrões, e problemas que não apresentam definição rigorosa. Esse é justamente o caso para o reconhecimento de voz. As redes neurais aprendem a desempenhar a função desejada através da apresentação de exemplos. Não é necessário, portanto, se descrever detalhadamente os critérios que separam uma palavra da outra. O sistema se torna mais robusto quando é apresentado ao maior número de exemplos possível, simulando o comportamento do cérebro.

Um modelo computacional de reconhecimento de voz pode ter diferentes objetivos:

A - Identificação de palavras: a mesma palavra pronunciada por pessoas diferentes deve ser tida como igual, ou seja, os padrões a serem reconhecidos são as palavras.

Aplicação possível: sistemas em que diferentes usuários dão os mesmos comandos vocais.

Exemplos: acionamento de equipamentos, auxilio para navegação em softwares para microcomputadores

B - Identificação de pessoas: há o reconhecimento da pessoa através da pronúncia de alguma palavra.

Aplicação possível: sistemas de segurança

Exemplos: controle de acesso em ambientes ou sites da internet

C - Identificação de pessoa/palavra: Uma amostra só é reconhecida quando uma determinada pessoa pronuncia uma determinada palavra. É a implementação mais simples. Aplicação possível: misto das anteriores, em versões menos dinâmicas.

Foi desenvolvido um software demonstrativo, RV-RN, que é um exemplo de uma aplicação prática do sistema. Realiza o reconhecimento como descrito pelo tipo A,

acima. Sua funcionalidade é movimentar um objeto na tela segundo comandos de voz de diferentes usuários. Os comandos são pré-definidos (“sobe”, “desce”, “esquerda” e “direita”). O sistema de reconhecimento por redes neurais permite que quando o sistema seja adaptado sempre que cometa erros de reconhecimento.

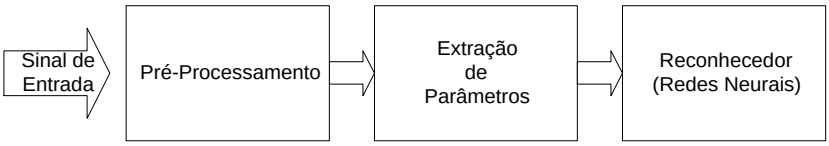
2. Materiais e Métodos

O material utilizado para o desenvolvimento do projeto foi um microcomputador equipado com alto-falantes, microfone e software MATLAB.

Com este equipamento foram colhidas amostras de voz de diversas pessoas. Após avaliação da qualidade destas amostras estas eram utilizadas no desenvolvimento e testes de performance do sistema.

3. Resultados

O sistema reconhecedor é dividido em três etapas conforme o diagrama de blocos abaixo. O detalhamento de cada etapa é apresentado nos tópicos seguintes.



3.1. Pré-Processamento

Características do sinal de entrada que refletem o ambiente de gravação e o canal de comunicação (assim como variações no estilo da fala), normalmente, atrapalham a tarefa do bloco reconhecedor. O sinal deve então ser parcialmente “limpo”, para que essas características sejam retiradas ou tenham sua influência na característica geral do sinal, diminuída. A tarefa de retirada de fatores estranhos ao sinal é feita por um estágio anterior ao bloco reconhecedor, o *pré-processamento*. Se as condições do ambiente são estacionárias ou suas variações podem ser determinadas, as características indesejadas podem ser removidas. Como mencionado na introdução deste documento, existe uma analogia entre este estágio funcional do projeto e o sistema de audição humano (ouvidos externo e médio).

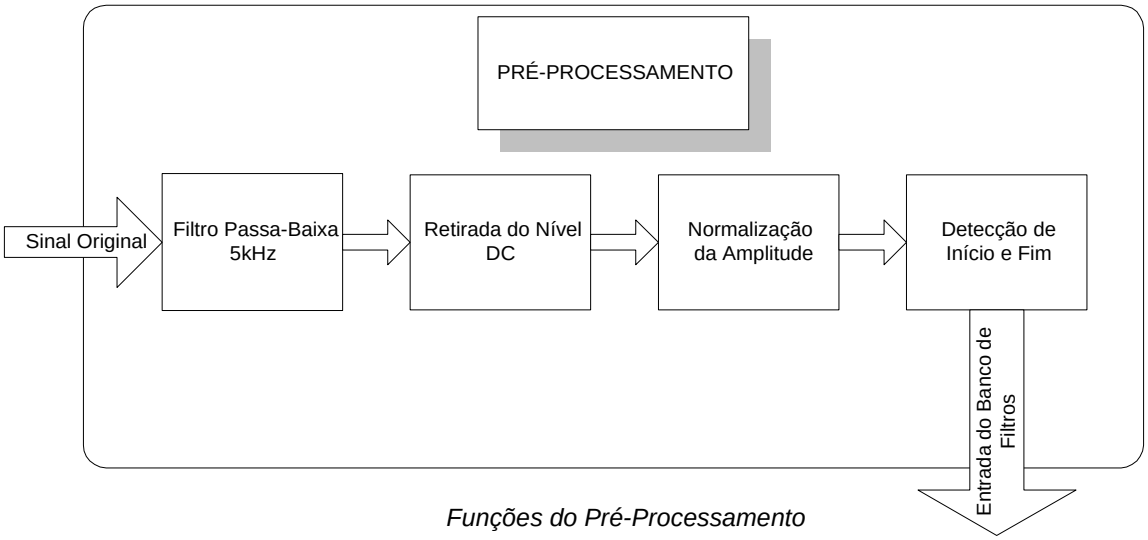
As funções que compõe este bloco e suas estratégias de funcionamento serão descritas adiante.

3.1.1. Metodologia

Características do sinal de voz original que normalmente prejudicam o reconhecimento são causadas por variações no ambiente de gravação como: ruídos de alta freqüência, variações no distanciamento do microfone, variação do nível de amplitude, variações de períodos de silêncio (antes do início do sinal e no seu fim).

Para anular a influência destas variações e facilitar o trabalho de reconhecimento, foram desenvolvidas 3 funções: filtragem anti-ruídos, retirada do nível DC, normalização da amplitude e corte dos períodos de silêncio no início e fim do sinal original. Estas funções, assim como o restante do sistema, foram desenvolvidas em ambiente Matlab. O diagrama de blocos abaixo explicita a seqüência em que estas funções ocorrem.

Para retirar os ruídos de alta freqüência, eventualmente, originados pelo ambiente, o sinal original é filtrado por um filtro passa-baixas de décima ordem e freqüência de corte de 5kHz. Isto não prejudica de nenhuma forma o sinal original pois as informações mais relevantes carregadas por um sinal de voz estão na faixa de 200 a 5600Hz. Continuando a analogia com o sistema de audição humano, este bloco desempenha a mesma função do ouvido médio, que atua como um filtro passa-baixas com atenuação de 15 dB/oct acima de 1 kHz.



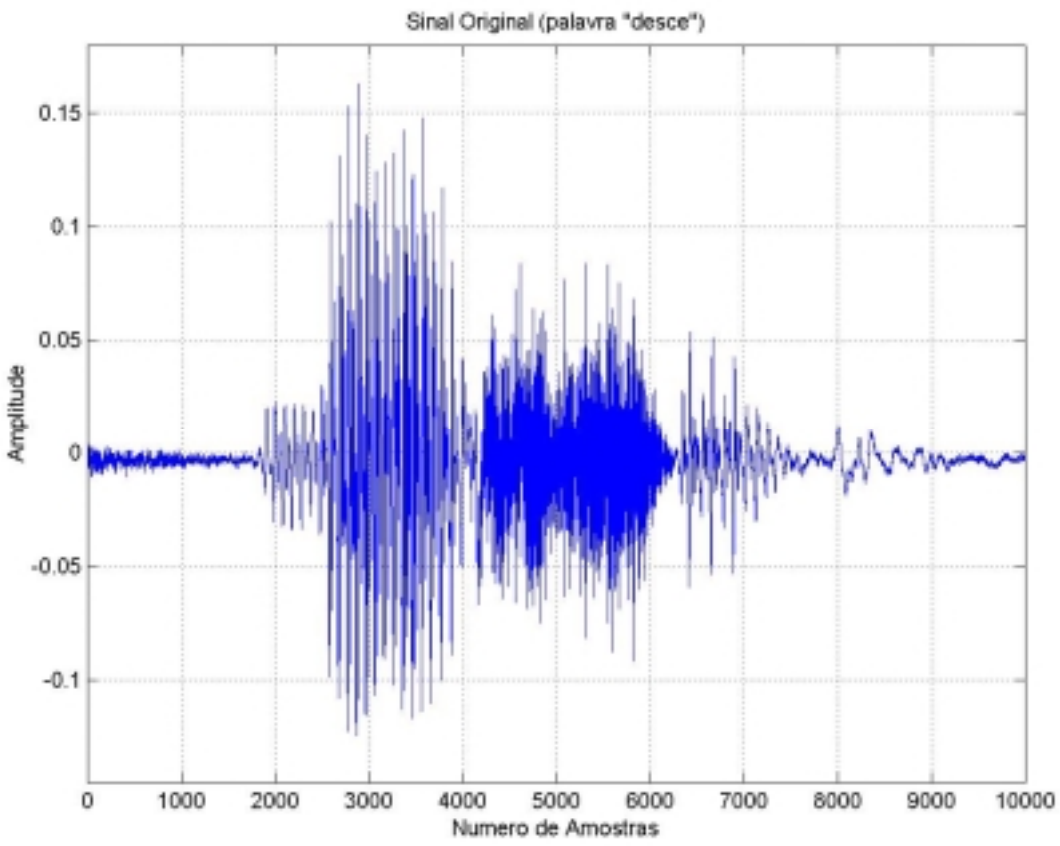
Funções do Pré-Processamento

Muitas vezes os sinais de voz apresentam uma componente contínua e para que uma comparação em valores absolutos possa ser realizada é necessário retirar esta componente DC. A eliminação deste nível coloca todos os sinais em relação a mesma referência. A função que realiza isto, "*retiradc*", calcula a média do sinal somando o valor de amplitude de cada amostra do sinal e em seguida divide esta soma pelo número de amostras do sinal. Uma vez calculada a média do sinal, ocorre um deslocamento em relação a esta média, isto é, o valor médio calculado é subtraído da amplitude de cada amostra do sinal. Todos sinais de entrada do sistema são gravados com 10.000 amostras a uma taxa de amostragem de 11.025Hz (taxa de amostragem padrão do Matlab).

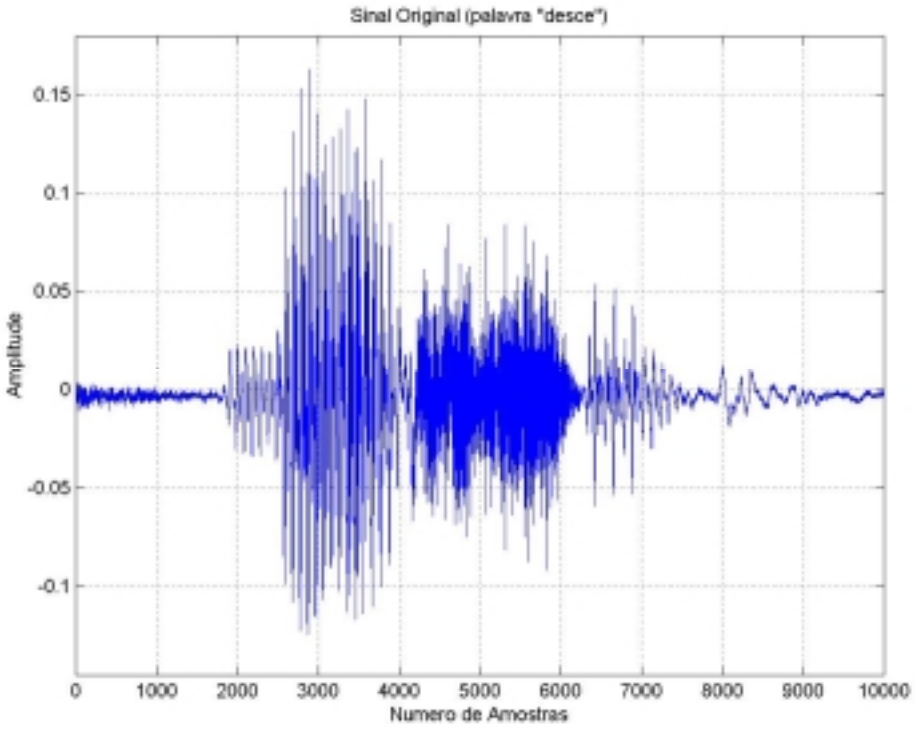
Para eliminar as que variações de amplitude do sinal, geradas na gravação por diferentes distanciamentos do microfone e diferentes intensidades de produção é necessário normalizar o sinal com relação a sua amplitude. Dessa forma a amplitude dos sinais fica limitada entre –1 e +1. Esta tarefa é realizada dividindo o valor de cada amostra do sinal pelo maior valor de amplitude do sinal. Esta estratégia mostrou-se mais eficaz quando comparada, por exemplo, com normalização logarítmica ou pela média.

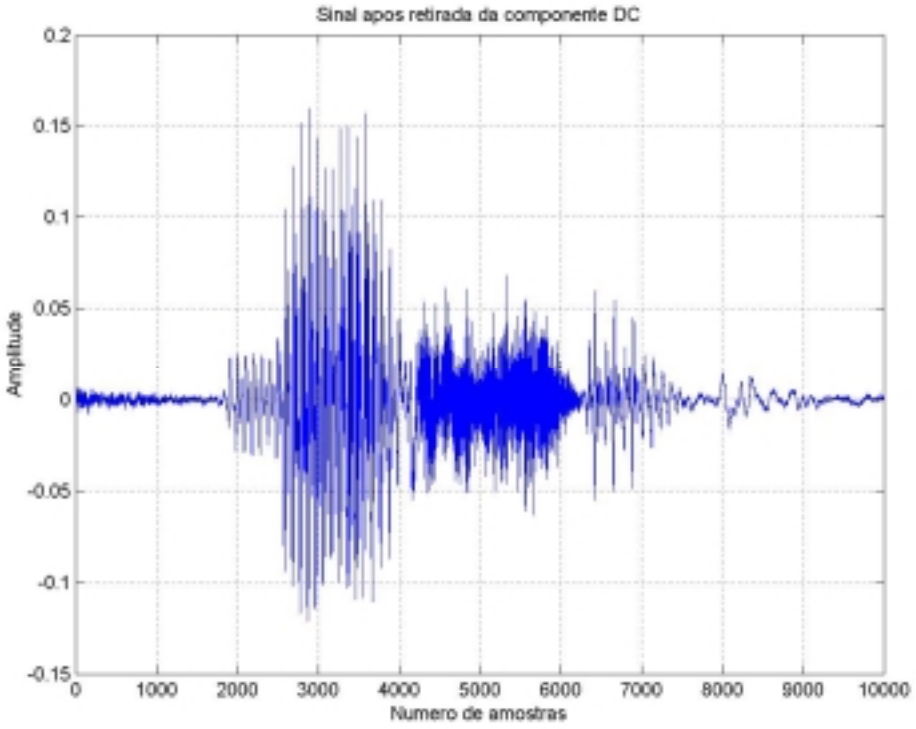
Os períodos de silêncio antes e após o sinal, além de não possuírem nenhuma informação valiosa para o reconhecimento, podem conter ruídos, sinais indesejados e a duração destes períodos pode ser muito variável. Portanto, é necessário retirar estes intervalos sem prejuízo para a parte que realmente contém informações relevantes. Para executar esta tarefa foi desenvolvida uma função, "corte", que elimina estes intervalos de acordo com parâmetros ajustáveis. A função "corte" recebe como entradas um valor de limiar (vlim), um valor médio de amplitude (vmlim) e um número de amostras (n). Com estes dados a rotina procura a primeira amostra cujo valor de amplitude ultrapassa o limiar indicado (esta procura parte do início para o fim). Este valor de limiar é experimental e pode ser ajustado de acordo com o desempenho apresentado pela função. Após localizada a amostra que ultrapassa o limiar, a rotina passa a calcular a média das amplitudes das n amostras subseqüentes até que este valor médio ultrapasse o valor de vmlim. Quando isso ocorre todas as amostras anteriores são descartadas. Esta estratégia é adotada para que a eliminação dos períodos sem informação seja otimizada, ou seja, se fosse usado apenas o valor de limiar existiria uma possibilidade de um ruído ser entendido como o início do sinal, acarretando na permanência de um período de silêncio. O mesmo método é utilizado para a retirada do final do sinal, após o término da fala do locutor. A rotina analisa o sinal de trás para frente, utilizando os mesmos parâmetros de entrada.

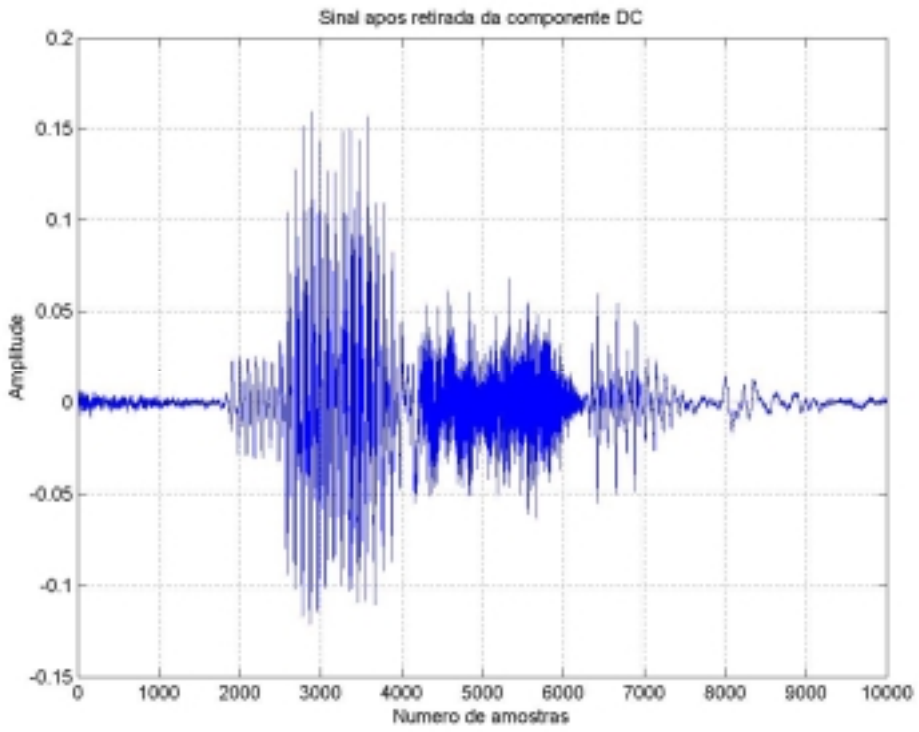
A seguir apresentamos, graficamente, os resultados do bloco de pré-processamento. Através dos gráficos apresentados ficam claras as alterações causadas por este bloco no sinal original e podemos verificar que o resultado esperado foi atingido. Para esta simulação foi utilizada a palavra "desce" como sinal original.

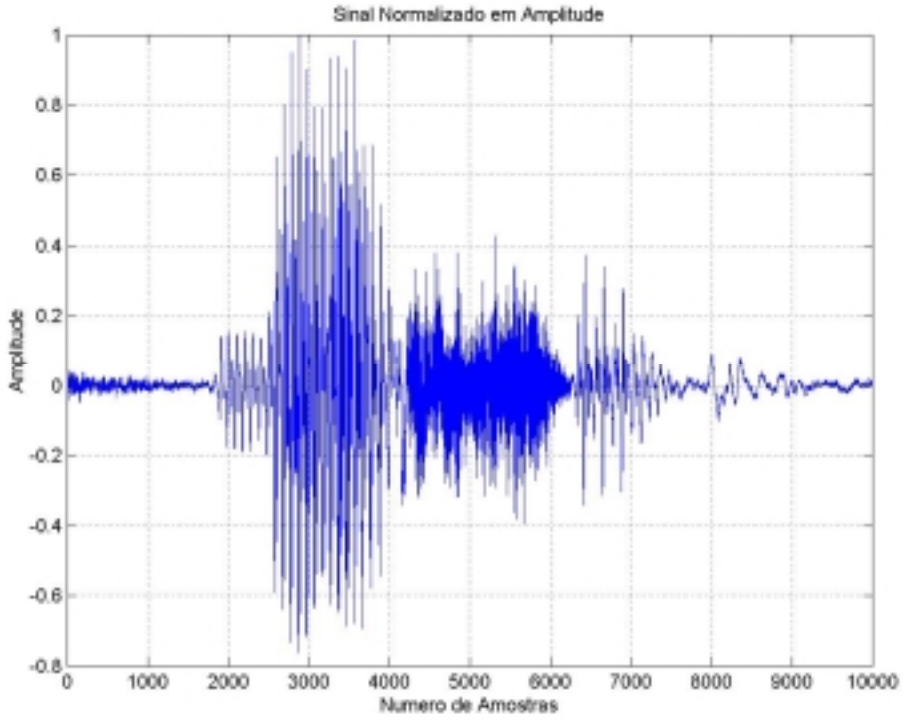


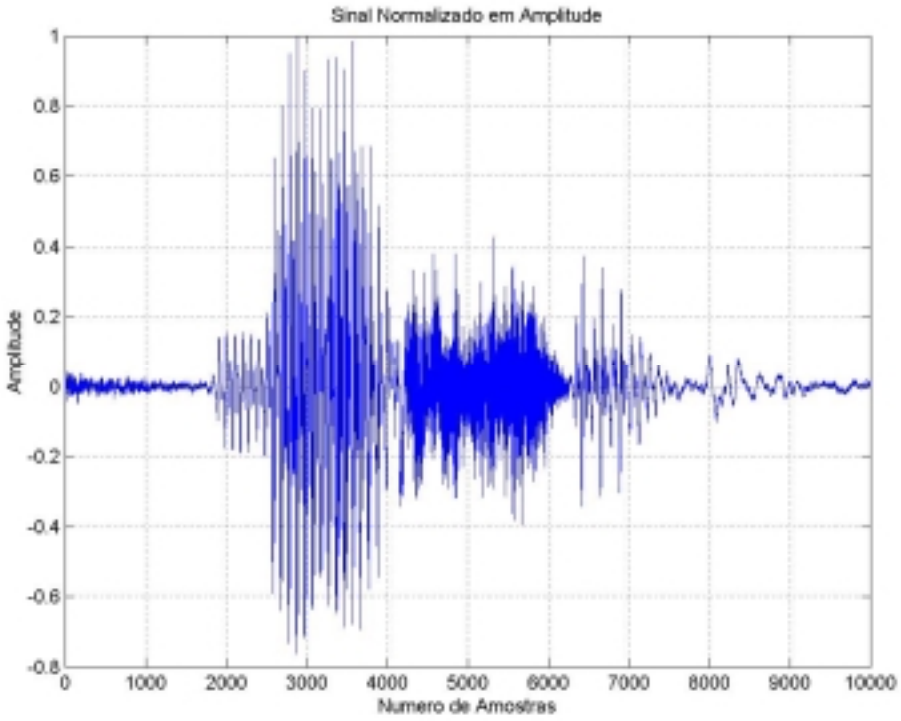
A plotagem do sinal com sua componente DC já retirada mostra o resultado da função de retirada do nível DC. Como neste caso o deslocamento do sinal é muito pequeno, fica mais fácil observar o resultado da aplicação da função quando reparamos no deslocamento da componente de maior amplitude. Nas páginas seguintes pode-se observar também o resultado da função de normalização da amplitude e em seguida o resultado da detecção de início e fim da palavra.

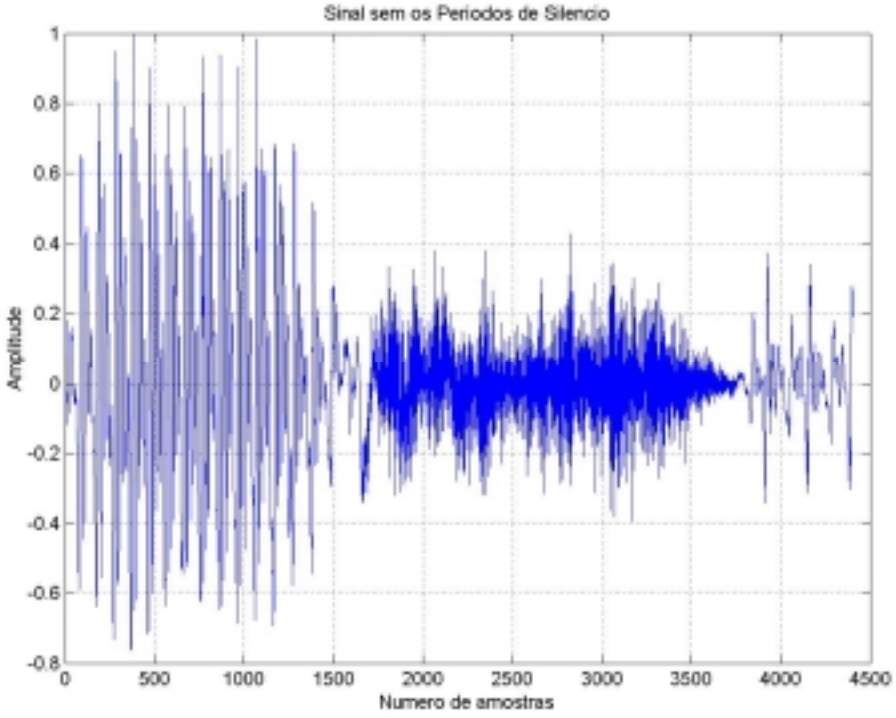

Sinal Original (palavra "desce")
Amplitude
Numero de Amostras
0.15
0.1
0.05
0
-0.05
-0.1
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000


Sinal apos retirada da componente DC
Amplitude
Numero de amostras
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000


Sinal apos retirada da componente DC
Amplitude
Numero de amostras
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000


Sinal Normalizado em Amplitude
Amplitude
Numero de Amostras
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000


Sinal Normalizado em Amplitude
Amplitude
Numero de Amostras
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000


Sinal sem os Periodos de Silencio
Amplitude
Numero de amostras
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
500
1000
1500
2000
2500
3000
3500
4000
4500

3.2. Extração de Parâmetros

Uma importante parte do processo de reconhecimento de voz diz respeito à escolha das características presentes em uma onda de voz que influenciarão a tomada de decisão pelo reconhecedor. Características baseadas no espectro do sinal de voz tem amplo uso em estruturas de reconhecimento de voz. Técnicas baseadas em FFT também tem servido para gerar padrões de voz que podem ser classificados. Neste projeto esta tarefa foi realizada por um modelo de Banco de Filtros. A extração dos parâmetros é fundamental para o sucesso do reconhecimento uma vez que permite a classificação de padrões através de algumas características do sinal.

No modelo de extração de parâmetros por banco de filtros, o sinal é passado por um conjunto de filtros passa-faixa. Este conjunto de filtros foi escolhido para cobrir a faixa de freqüência mais importante em um sinal de voz, 50 – 4800Hz. Mais uma vez a analogia com o sistema de audição humano foi importante, levando a uma distribuição das faixas centrais dos filtros segundo a escala de Mel ou Bark.

3.2.1. Metodologia

Após sucessivos testes, estudos e analises de desempenho das redes, determinou-se o uso da energia do sinal como entrada do bloco reconhecedor. Para tal, o sinal de saída de cada filtro do banco foi dividido em trechos menores e então a energia de cada um destes trechos foi calculada. Este método permitiu a preservação da característica tempo-frequência do sinal, que é muito importante para a classificação dos padrões.

O banco de filtros é formado por 19 filtros passa-faixa com características de banda de passagem e freqüência central projetadas segundo a escala de Mel ou Bark. A escala de Bark vai de 1 até 24 Barks, correspondendo às primeiras 24 bandas críticas da audição. Calculando-se os coeficientes de Mel pela aproximação apresentada a seguir é possível chegar-se até a escala publicada por Bark, apresentada na tabela 1.

$$m(f) = 2595 \log\left(1 + \frac{f}{700}\right)$$

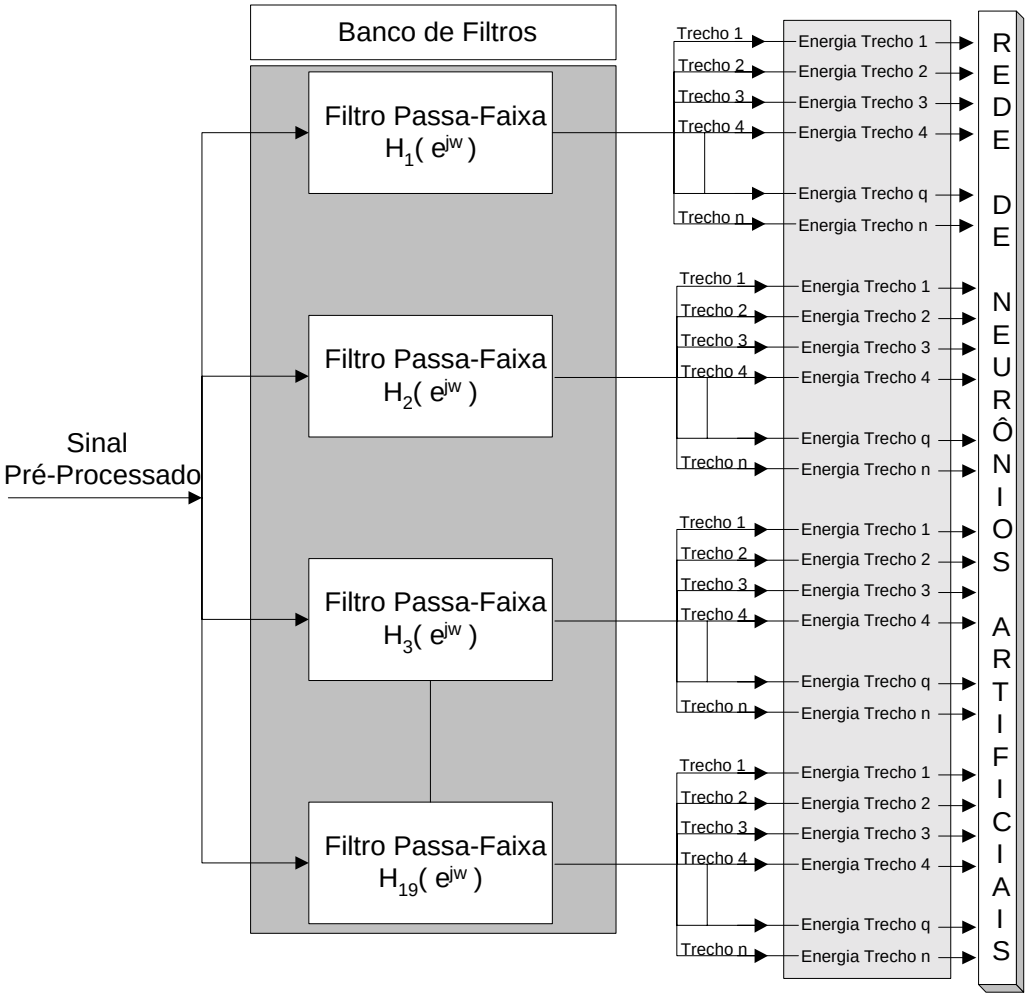
O gráfico abaixo é o registro da resposta em freqüência do banco de filtros projetado. Neste gráfico pode ser observada, em detalhes, a influência da escala de Mel na distribuição das freqüências centrais e das bandas de passagem. Todos os filtros são de segunda ordem e foram projetados através da função *butter* do Matlab (depois de calculadas a freqüência central e a banda de cada um).

Tabela 1

Freqüências centrais e bandas de passagem do banco de filtros	
	0
Filtro 1	**f_{c1} = 50 Hz**
	100
Filtro 2	**f_{c2} = 150 Hz**
	200
Filtro 3	**f_{c3} = 250 Hz**
	300
Filtro 4	**f_{c4} = 350 Hz**
	400
Filtro 5	**f_{c5} = 450 Hz**
	510
Filtro 6	**f_{c6} = 570 Hz**
	630
Filtro 7	**f_{c7} = 700 Hz**
	770
Filtro 8	**f_{c8} = 840 Hz**
	920
Filtro 9	**f_{c9} = 1000 Hz**
	1080
Filtro 10	**f_{c10} = 1170 Hz**
	1270
Filtro 11	**f_{c11} = 1370 Hz**
	1480
Filtro 12	**f_{c12} = 1600 Hz**
	1720
Filtro 13	**f_{c13} = 1850 Hz**
	2000
Filtro 14	**f_{c14} = 2150 Hz**
	2320
Filtro 15	**f_{c15} = 2500 Hz**
	2700
Filtro 16	**f_{c16} = 2900 Hz**
	3150
Filtro 17	**f_{c17} = 3400 Hz**
	3700
Filtro 18	**f_{c18} = 4000 Hz**
	4400
Filtro 19	**f_{c19} = 4800 Hz**
	5800

Na tabela acima verificamos a freqüência central e a banda de cada um dos 19 filtros do banco. Os valores imediatamente acima e abaixo da freqüência central são os limites da banda de passagem, ou seja, valores onde há atenuação de 2dB no sinal.

O diagrama de blocos apresentado a seguir ilustra a técnica de extração de parâmetros utilizada no projeto:

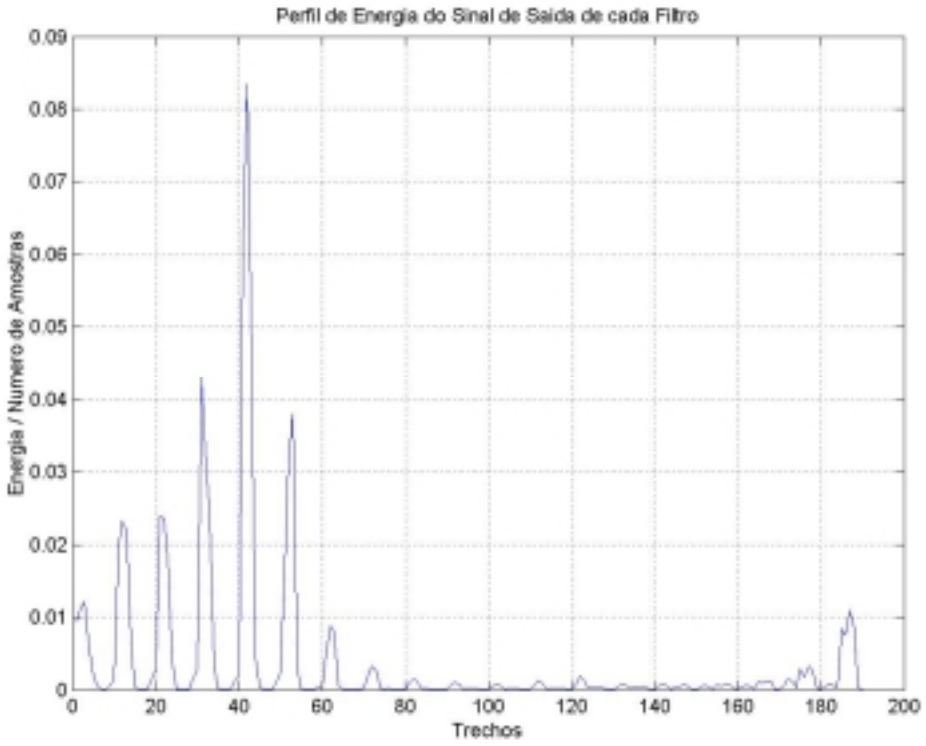


A saída de cada filtro é dividida em trechos menores para preservação da característica tempo-freqüência. O número de trechos é determinado a partir do número de entradas da rede neural (número de entradas da rede / 19). Para que tenhamos sempre um número inteiro de trechos, a única restrição para o número de entradas da rede é que este seja múltiplo de 19. O sinal pré-processado atravessa o banco, a saída de cada filtro é dividida em trechos de acordo com o número de entradas desejado para a rede, em seguida, calcula-se a energia de cada trecho. Este cálculo de energia é feito segundo a relação de Parseval:

$$\frac{1}{N}\sum_{n=-\infty}^{+\infty}\left|x[n]\right|^2$$

A função que realiza a divisão das respostas dos filtros em trechos e o cálculo da energia de cada trecho recebe como parâmetro apenas o número de entradas da rede de neurônios. O resultado da etapa de extração de parâmetros pode ser visto no gráfico

abaixo, onde foi informado ao sistema que a rede teria 190 entradas. Neste gráfico observamos a energia associada a cada trecho de saída dos filtros (190 trechos).



Para simplificar a utilização foi elaborada uma rotina que agrupa as funções de pré-processamento e extração de parâmetros. As entradas desta função são: o número de entradas da rede de neurônios e os limiares da função de corte. As saídas geradas pela função servem diretamente como entradas da rede neural.

É importante o acompanhamento dos gráficos apresentados, tanto do pré-processamento quanto da extração de parâmetros para que seja possível avaliar se o desempenho destas etapas está satisfatório.

3.3. Redes Neurais

Nesse projeto, as redes de neurônios artificiais, ou redes neurais, são responsáveis por realizar o reconhecimento da voz em si. As redes neurais são modelos computacionais que resolvem problemas que não apresentam uma especificação rigorosa, especialmente do tipo de reconhecimento de padrões, como é o caso desse projeto. Embora seja absolutamente intuitiva para os humanos, a tarefa de reconhecimento de voz é muito difícil de ser implementada computacionalmente, pois jamais a mesma palavra é pronunciada igualmente duas vezes.

As redes neurais são sistemas baseados no comportamento dos neurônios reais, no que se refere às suas interconexões e tratamento de sinais de entrada para geração do sinal de saída. As redes são constituídas de vários neurônios interconectados, que na realidade são células de processamento com parâmetros ajustáveis. Elas efetivamente

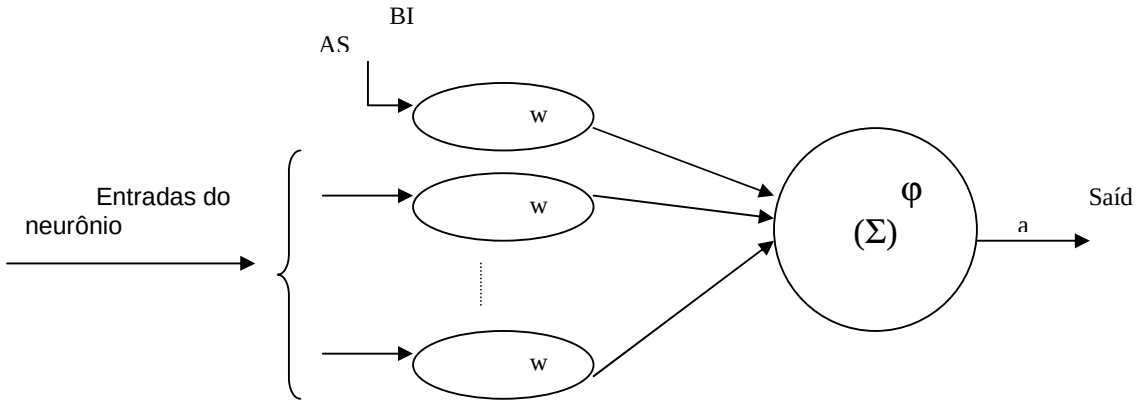
aprendem a realizar a função desejada através da realização de seu treinamento. Nessa etapa se apresenta para a rede exemplos do funcionamento que se quer que ela tenha.

É exatamente essa a funcionalidade de nosso projeto e do programa demonstrativo, RV-RN. Através de exemplos gravados pelo usuário, que constituem a base de treinamento do sistema ou o banco de dados do sistema, a rede neural é treinada para executar a função de reconhecimento de padrões – os padrões dos comandos de voz pré-definidos.

3.3.1. Modelo das Redes Neurais

O tipo de rede neural utilizado foi o MLP – "Multi Layer Perceptron", que consiste na composição de uma rede conectando-se vários neurônios do tipo Perceptron.

Cada perceptron (neurônio artificial) modela o comportamento dos neurônios naturais. Possui pesos sinápticos (wi) que simulam suas conexões aos neurônios de entrada e uma função de transferência, que simula seu comportamento entrada-saída. A saída é calculada com base em suas entradas, que na rede são as saídas de outros neurônios. Segue o modelo do perceptron:

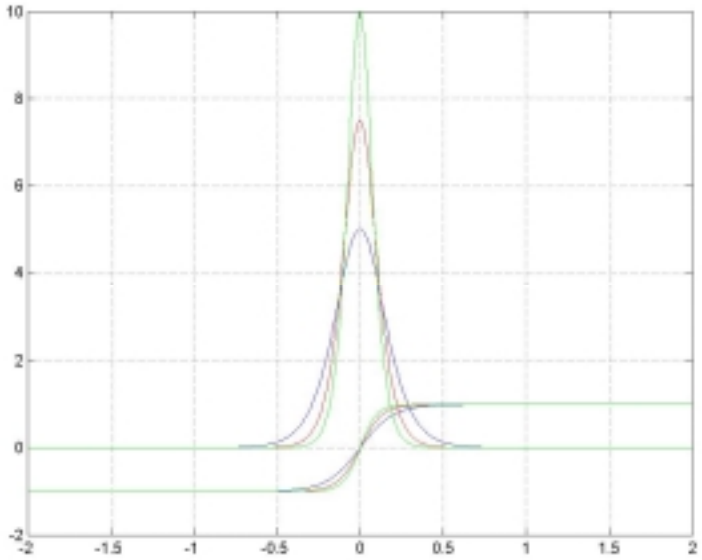


O perceptron é, portanto, uma célula simples de processamento com parâmetros ajustáveis. Toma decisões baseadas em suas entradas e consegue resolver problemas simples, como por exemplo, classificação de padrões com separabilidade linear.

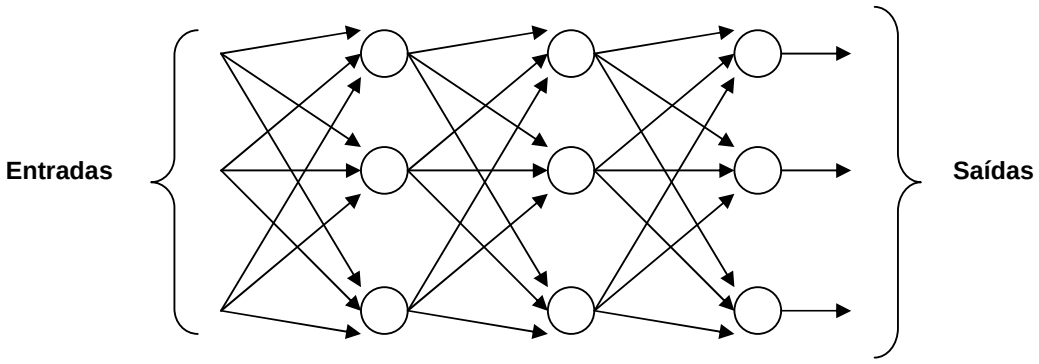
Cada neurônio calcula a somatória dos pesos sinápticos pelos respectivos sinais de entrada, além do sinal de Bias, constante igual a +1. Após calculado esse valor (sinal de ativação, vk), é calculado o sinal de saída, através da função de ativação (φ). A função de ativação utilizada foi a sigmóide, que tem a expressão abaixo

$$\varphi(vk) = \frac{1}{1+ \exp(-ks \times vk)}$$

É uma função ímpar e limitada entre –1 e +1. Note que ela é não-linear, o que é uma característica muito importante, pois caso contrário, a rede toda poderia acabar tendo a funcionalidade de um simples perceptron isolado. Pode-se obter variações da função variando o valor da constante da sigmóide, ks. O gráfico abaixo mostra três formas de sigmóide e suas respectivas derivadas (as sigmóides são os sinais de menor amplitude).



A estrutura típica de MLP contém uma camada de neurônios de entrada e uma camada de neurônios de saída além de camadas escondidas. Essas camadas escondidas possibilitam que a rede aprenda tarefas complexas, extraindo parâmetros progressivamente mais significativos a partir dos parâmetros de entrada. O sinal se propaga através da rede em sentido único, camada a camada, isto é, não há feedback. Cada neurônio tem como entrada as saídas de todos os neurônios da camada anterior.



3.3.2. Treinamento da Rede

As redes neurais podem desempenhar uma infinidade de funções complexas, mas ao contrário de mecanismos usuais de programação a funcionalidade do algoritmo não é descrita explicitamente. As redes devem ser treinadas para adquirir a funcionalidade desejada, através da execução de Treinamento Supervisionado. Através de exemplos de funcionamento ensina-se às redes a desempenhar o processamento desejado.

Este procedimento submete as redes a uma seqüência de amostras de aprendizado em que se sabe qual o resultado desejado que a rede deverá apresentar. Aplica à rede um conjunto de entrada e caso haja erro em sua saída, corrige seus parâmetros, de modo que ela apresente o comportamento apropriado. A rede, portanto, se auto-ajusta através de exemplos, para aprender a realizar a função desejada.

O algoritmo de treinamento utilizado foi o "Back Error Propagation", baseado no aprendizado por erro. Este algoritmo aplica um conjunto de entradas a uma rede neural, e verifica se há erro de classificação, isto é, se a saída da rede não corresponde à saída esperada para as entradas aplicadas (caso o padrão apontado pela rede seja diferente do padrão esperado para o dado vetor de entradas). Se houver erro, o mesmo é propagado para trás, em direção à primeira camada da rede, camada a camada através das conexões sinápticas.

Os algoritmos desenvolvidos serão descritos mais adiante.

Assim como o desenvolvimento do software demonstrativo RV-RN, a implementação das redes neurais foi realizada utilizando Matlab. Sua funcionalidade é obtida utilizando-se dois sub-programas, um para execução do processamento da rede neural em si (REDE), e outro para realização do treinamento da rede (TREINA). Os próximos itens descrevem esses sistemas detalhadamente.

O conhecimento adquirido sobre redes neurais aplicadas a reconhecimento de voz pode ser utilizado para funções diversas. O sistema demonstrativo elaborado, RV-RN, possui a funcionalidade de reconhecer comandos de voz pré-definidos, independentemente do usuário. Os comandos ("sobe", "desce", "esquerda" e "direita") devem ser reconhecidos e interpretados pelo sistema. Cada um deles deve corresponder a execução de uma ação, mesmo se pronunciados por locutores diferentes. Dessa forma não é necessário que o sistema seja capaz de realizar a diferenciação dos usuários, o que permite reduzir o tamanho da rede, e conseqüentemente seu tempo de processamento e de treinamento, permitindo-nos focar nos objetivos da funcionalidade proposta e orientar as melhorias de performance nesse sentido.

A identificação de usuários é uma característica essencial em outros tipos de sistemas, como os de acesso de segurança, por exemplo. No entanto não foi realizado esforço para o desenvolvimento dessa linha de funcionalidade e de outras que não a proposta no escopo inicial, pois não desejávamos ter atrasos no projeto devido ao alargamento do escopo.

Embora não faça parte do escopo do projeto, a flexibilidade do sistema permite também que a funcionalidade do sistema demonstrativo seja facilmente adaptada para executar outras funcionalidades, conforme será descrito a seguir.

Outra premissa importante para a elaboração dos programas foi com relação ao aprendizado do sistema a partir de seus erros. Quando cometer algum erro de reconhecimento, isto é, quando realizar uma ação não-correspondente ao comando dado, o usuário pode ensinar ao sistema que errou e como corrigir seu erro. Das próximas vezes o erro não se repetirá porque p sistema aprendeu com o erro. Dessa forma o RV-RN pode adquirir funcionamento mais robusto, em tempo de execução.

Uma das metas ao se desenvolver essas sub-rotinas era se garantir a flexibilidade das mesmas, portanto a metodologia de desenvolvimento dos dois programas, REDE e TREINA, foi orientada nessa direção. Ambos têm um grande número de parâmetros que devem ser ajustados (otimizados) para a utilização no software. Muitos deles influem no desenvolvimento do código, mas durante a etapa de elaboração do código a maioria ainda não estava definida e a adaptação do código caso a caso, de modo que fosse possível testar-se várias configurações, teria sido extremamente onerosa.

O procedimento da definição de parâmetros não buscava apenas obter funcionamento apropriado, mas também alguma otimização. Portanto, para fazê-lo de modo adequado, foi necessário realizar-se muitos testes, obtendo resultados de performance para o maior número possível de combinações de valores que os diversos parâmetros poderiam assumir. Isso só foi possível devido ao caráter flexível dos programas de redes neurais que permitiam que os testes que envolviam grande variação das características das redes, inclusive mudança de arquitetura, fossem executados de modo muito prático e sem nenhuma alteração de código.

No decorrer do desenvolvimento do projeto nos deparamos com diversas decisões que orientariam o projeto entre dois caminhos absolutamente distintos. Para ilustrar melhor essas decisões, sua descrição será feita junto ao detalhamento dos resultados obtidos, no próximo item do relatório. Dessa forma, explicitamos a solução adotada, e a comparamos às outras opções, justificando sua escolha.

Em geral, as decisões com relação à definição dos rumos do desenvolvimento do projeto foram realizadas utilizando-se uma ou mais das ações abaixo:

- Criação de uma ou mais alternativas para resolução de algum problema
- Apresentação das alternativas ao professor orientador
- Discussão com o professor a respeito das alternativas propostas e de novas alternativas propostas por ele
- Busca de embasamento teórico em referências bibliográficas
- Realização de testes comparativos com todas as alternativas, quando necessário, isto é, quando o bom-senso não era capaz de eliminar algumas delas.

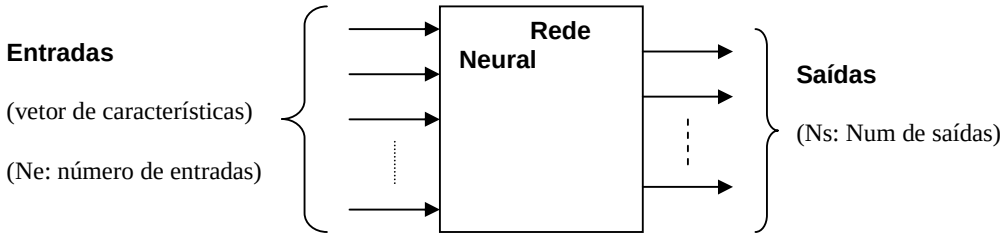
3.3.3. O programa de Redes Neurais (REDE)

Este programa implementa o funcionamento de uma rede neural MLP, isto é, dado um conjunto de entradas, que é um vetor de características extraídas da amostra de voz a ser analisada, são calculadas as saídas da rede, que classificam o sinal de entrada. O reconhecimento de cada palavra analisada, portanto, é realizado executando-se essa

rotina a partir do vetor de características extraídas do sinal pela etapa anterior (pré-processamento + extração de parâmetros).

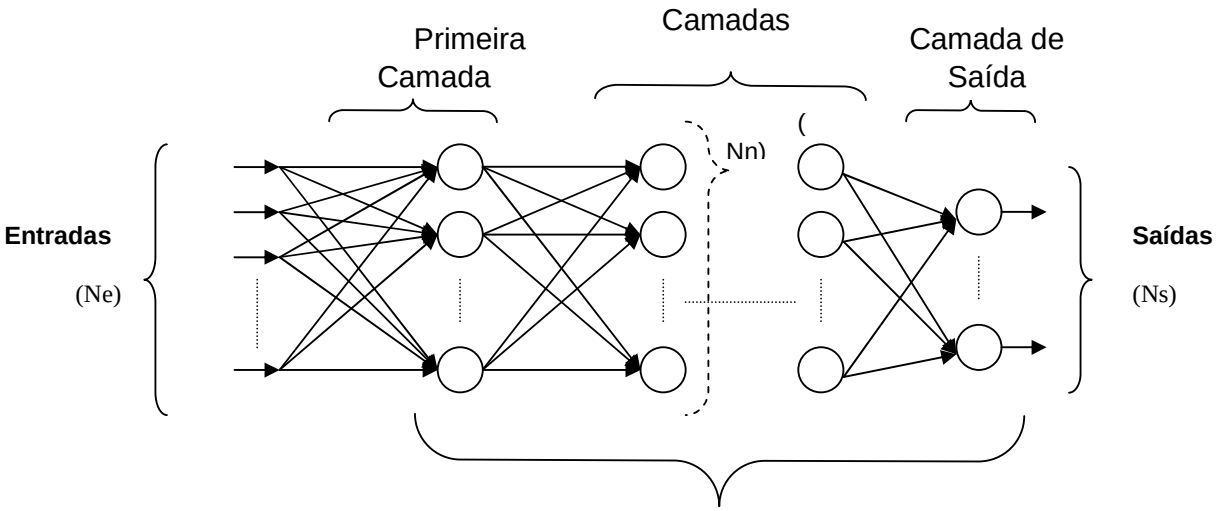
O programa é totalmente flexível e pode simular uma infinidade de arquiteturas de redes MLP. Essa flexibilidade, característica tida como essencial nesse projeto, é obtida através da utilização de muitos parâmetros. Tudo o que se pode variar na rede está na forma de parâmetros. Mostraremos agora a estrutura da rede neural implementada e todos os parâmetros do programa.

Modelo Macroscópico da Rede Neural:



O número de saídas (Ns) e o número de entradas (Ne) podem ser alterados.

Modelo Microscópico da Rede Neural:



O vetor de características é aplicado à primeira de seus neurônios geram suas respectivas saídas, que por sua vez são as entradas da próxima camada. O processo ocorre até que se atinja a camada de saída.

Segue lista de todos os parâmetros do programa:

W: Pesos Sinápticos da rede a ser simulada, de onde o programa extrai:

Ne: Número de Entradas da rede neural

Nn: Número de Neurônios por camada

Nc: Número de Camadas da rede

Ns: Número de Saídas da rede

Xe: Vetor de entrada da rede a ser simulada

Ks: Constante da sigmóide utilizada nos neurônios

Inicialmente o desenvolvimento do projeto foi orientado para que houvesse uma rede neural para cada uma das palavras que constituem a base de dados do sistema, isto é, uma rede para cada usuário para cada um dos quatro comandos. Após execução de vários testes essa abordagem foi abandonada porque necessitava de muito tempo para executar os cálculos referentes à rede neural e para executar o treinamento da rede. Além disso, o re-treinamento das redes era muito lento a cada vez que uma nova amostra fosse incorporada à base de dados da rede, uma vez que deveriam ser re-treinadas muitas redes. Além de onerosa, essa solução não apresentava melhora aparente nos resultados obtidos com relação à solução definitiva.

Posteriormente a solução adotada foi utilizar apenas uma rede. Pensou-se então que a utilização de uma só rede neural poderia dificultar muito o trabalho de classificação feito pelas redes, pois teria que criar padrões que independem do usuário. Isto é, a saída ativa da rede apenas depende do comando falado, e não do usuário. O padrão então deveria englobar diferentes usuários. Para evitar esse agravante facilitando o trabalho das redes, adotou-se então um conjunto de saídas para cada usuário, isto é, havia uma saída para cada comando para cada usuário cadastrado no sistema. Como eram quatro comandos (“sobe”, “desce”, “esquerda” e “direita”) deveria haver quatro saídas para cada usuário. Como resultado, o treinamento da rede apresentava melhor tempo de treinamento, mas ainda era demasiado demorado. Além disso, a preocupação com a dificuldade de reconhecimento não se mostrou necessária, pois as redes foram capazes de executar a classificação utilizando-se apenas quatro saídas. Melhores resultados foram obtidos com essa configuração.

O sistema RV-RN, portanto, conta com apenas uma rede neural, que possui, de acordo com as premissas estabelecidas no item 2.1, cinco saídas. Cada saída corresponde a uma palavra, não diferenciando o usuário.

Abaixo, vemos a descrição da semântica das saídas:

- Saída 1: comando “Sobe”
- Saída 2: comando “Desce”
- Saída 3: comando “Esquerda”
- Saída 4: comando “Direita”
- Saída 5: palavras que devem ser rejeitadas

Seus valores podem variar entre –1 e +1, sendo que +1 corresponde a identificação e –1 a rejeição. Dessa forma, se dado um vetor de entrada, alguma das saídas estiver próxima de +1, o sinal será interpretado pela rede como sendo aquela palavra. Exemplo:

Saída 1: -0,87

Saída 2: -0,92

Saída 3: +0,91

Saída 4: -0,57

Saída 5: -0,77

Nesse caso, o sistema reconhecerá a amostra testada como a palavra correspondente á saída 3, ou seja, o comando "esquerda".

No entanto nem sempre a decisão será tão fácil quanto foi nesse exemplo. Há por exemplo casos em que nenhuma das amostras estará suficientemente próxima de +1 para que possamos afirmar que se trata, com certeza, de algum dos comandos pré-definidos. Para isso foi estabelecido um limiar mínimo de aceitação, de +0,5, isto é, se nenhuma das saídas apresentar saída superior a esse valor, o sistema não reconhecerá a palavra e não executará nenhuma ação. Por outro lado, pode acontecer de que mais de uma saída esteja próxima de +1. Nesse caso o sistema também não tomará nenhuma ação.

Lembre-se de que, no RV-RN, para esses casos em que o sistema errou ou não foi capaz de tomar uma decisão, o usuário pode ensiná-lo, baseando-se em seu erro. Nas próximas vezes que essas palavras forem ditas não haverá mais problemas de reconhecimento, pois o sistema efetivamente aprendeu a desempenhar o reconhecimento da forma determinada pelos exemplos apresentados a ele.

Note que o número de entradas(Ne) geralmente é maior que o número de neurônios por camada (Nn). O compromisso entre esses números é muito importante, pois por um lado podemos ter uma rede muito pequena que não é capaz de realizar o reconhecimento apropriadamente, e por outro podemos ter uma rede muito grande (com número de neurônios por camada igual ao número de entradas) com tempos de processamento e treinamento proibitivamente altos.

Além disso, o número de entradas da rede também é um parâmetro a ser otimizado. Pode-se preparar o sistema para trabalhar com qualquer número de entradas para a rede neural. E novamente há aqui um compromisso entre tempo e qualidade de processamento. Essa preocupação também existe no dimensionamento da rede, ao se determinar o número de camadas. Um dos principais objetivos ao se desenvolver o programa de redes neurais, portanto, foi justamente a questão de se obter bom funcionamento e tempos aceitáveis. O sistema como um todo deve ter tempo apropriado de treinamento e de execução, sem deixar de funcionar apropriadamente. Obviamente é impossível dizer que o sistema foi otimizado, pois, por mais testes que se faça, é impossível se prever como serão as amostras encontradas no futuro, e apenas pode ser obtida uma otimização para as amostras utilizadas nos testes, e não para amostras genéricas. No entanto, através de uma série de testes chegamos a valores de parâmetros que apresentam um bom comportamento balanceado entre tempo e qualidade de resultado, atendendo a nossas necessidades. Os valores dos parâmetros utilizados no sistema demonstrativo são mostrados abaixo. Os dados sobre confiabilidade do sistema podem ser vistos no conteúdo desenvolvido pelo aluno Gabriel.

3.3.4. O programa de Treinamento das Redes Neurais (TREINA)

É durante a etapa de treinamento que as redes neurais assimilam a funcionalidade que desejamos que tenha. Através da apresentação de amostras de voz e das saídas esperadas, os pesos sinápticos são adaptados de modo que a rede execute a função desejada, ao menos para os conjuntos de entradas apresentados. A etapa de treinamento é, portanto, extremamente importante.

Foi utilizado o algoritmo de "back error propagation", que é amplamente utilizado para redes MLP. Resumidamente, o algoritmo faz o seguinte:

- Aplica um dos conjuntos de treinamento à rede,

- Compara sua saída a saída esperada, se não houve erro, não faz nada e aplica a próxima entrada,

- Se houve erro, o mesmo é propagado para trás, camada a camada até a camada de entrada, através dos pesos sinápticos e utilizando-se a derivada da função de transferência dos neurônios.

- O processo é repetido, aplicando-se repetitivamente todo o conjunto de treinamento, variando-se a ordem de aplicação das amostras.

- Quando parar? Quando a rede apresentar comportamento apropriado para todos os conjuntos de entradas. O critério de parada é que o erro em todas as saídas, para todos os conjuntos, deve ser inferior a um valor pré-determinado empiricamente (Em)

- Há outro critério de parada: número de iterações. Quando o número de iterações máximo (Nit) é atingido, o treinamento é interrompido sem sucesso.

O conjunto de treinamento é constituído pelas amostras de voz gravadas para ensinar a rede. É a base de dados. Usualmente são gravadas duas amostras de cada comando por cada usuário. No decorrer da utilização do sistema RV-RN, o usuário acrescenta novas amostras à base de dados à medida que a rede realiza erros de reconhecimento. Quando a rede é treinada são passados como parâmetros todas as amostras de treinamento e as respectivas saídas. Exemplo: se uma amostra corresponde ao comando "desce", a saída esperada passada para o programa de treinamento será:

Saída 1: -1

Saída 2: +1

Saída 3: -1

Saída 4: -1

Saída 5: -1

O programa de Treinamento de Redes Neurais também é totalmente flexível. Pode gerar qualquer estrutura de rede MLP, baseado ou não em uma rede já previamente

existente. Ao se treinar uma nova rede neural, ou ao se re-treinar uma pré-existente, deve-se definir os seguintes parâmetros:

Parâmetro	Descrição	Valor utilizado
Ne	Número de Entradas da rede neural	190
Ns	Número de Saídas da rede	5
Nn	Número de Neurônios por camada	40
Nc	Número de Camadas da rede	4
Xt	Vetores de entrada, conjuntos de treinamento	-
Yt	Saídas esperadas para cada um dos vetores do conjunto de treinamento	-
Ks	Constante da sigmóide utilizada nos neurônios	1
Ka	Constante de aprendizado inicial	0,2
Em	Erro mínimo aceito para término do treinamento	0,1
Nit	Número máximo de iterações permitidas	1000
Wi	Pesos Sinápticos da rede a ser re-treinada, utilizado como valor inicial	-

Note que o treinamento é uma etapa muito sutil e apresenta possibilidade de não ser bem sucedido. Isto pode ocorrer caso as amostras de entrada (conjunto de treinamento) sejam de baixa qualidade. O que seria baixa qualidade? Por exemplo, se duas amostras da mesma palavra, do mesmo locutor, que devem ser interpretadas igualmente pela rede, forem muito diferentes. Grandes diferenças de velocidade e cadência da pronúncia, ou timbre ou ainda corte inapropriado na gravação podem representar grandes obstáculos ao algoritmo de treinamento, tornando-o muito extenso, ou até impossível.

Um fator muito importante a ser analisado nesse programa é seu tempo de execução. Como já foi dito o treinamento é a parte mais importante para a implementação das redes neurais, e também a mais demorada. Além disso, é sensível, e algumas vezes não pode ser completado com sucesso dentro do número máximo de iterações permitido. Por outro lado, se o número de iterações for aumentado, o tempo de treinamento se torna excessivamente longo. Durante o desenvolvimento tomou-se cuidado com relação a esse aspecto, e o resultado final em geral tem sido bom, com altos índices de treinamentos bem-sucedidos e com tempo aceitável de execução.

Visando melhorar o tempo de execução desse programa foi utilizado o recurso de variação da constante de aprendizado (ka). A constante de aprendizado é um valor multiplicativo aplicado à correção dos pesos sinápticos durante o treinamento. Quanto maior for o seu valor, maior será a variação dos pesos a cada iteração e maior a velocidade do treinamento. Por outro lado, quanto menor for o seu valor, maior será a precisão do algoritmo e menor a velocidade de treinamento. Há aqui um equilíbrio entre velocidade de treinamento e precisão. O recurso implementado visa obter o máximo dos aspectos positivos de cada lado da balança, isto é, obter velocidade e precisão. A constante de aprendizado é iniciada com um valor alto (ka) para que as etapas iniciais do treinamento obtenham avanços significativos em direção ao resultado final, isto é, para que a evolução seja mais rápida. No entanto, no decorrer do treinamento, o valor da constante é reduzido, visando obter-se a precisão necessária ao final do processo. A redução de seu valor é baseada na variação total de todos os pesos sinápticos entre uma iteração e outra, portanto ela reflete qual o tipo de ação está sendo tomada na rede: correção com altos valores ou correção com precisão.

No que tange ao tempo de execução do programa de treinamento deve-se fazer mais uma consideração: o sistema poderia apresentar performance significativamente melhor caso fosse utilizada alguma linguagem compilada, por exemplo, linguagem C. Note que isto foge ao escopo proposto pelo projeto. O programa de redes neurais também poderia ter sua performance melhorada dessa forma, o que seria importante caso se pensasse em uma aplicação de tempo real, que não é o caso.

A adição de amostras à base de dados da rede é uma funcionalidade implementada intrinsecamente no programa. Ao se adicionar uma nova palavra ao vocabulário, basta executar o treinamento da rede tendo como valores iniciais dos pesos sinápticos, os valores previamente existentes. Pensou-se que isso poderia acarretar alguma espécie de dificuldade, mas não houve motivos para preocupação.

3.3.5. O Software Demonstrativo

Também em Matlab foi desenvolvido um software para realizar a demonstração dos resultados obtidos com o sistema de reconhecimento

A partir da tela inicial pode-se:

- Abrir o módulo de gravação
- Executar o treinamento da rede neural
- Abrir o módulo de teste
- Resetar a base de dados do sistema

A plataforma de demonstração é constituída dos módulos descritos a seguir.

3.3.5.1. Módulo de Gravação

Nesse módulo são gravadas as amostras de voz que dão à rede seu treinamento inicial, ou seja, são gravados os conjuntos de treinamento iniciais. Pode-se criar novos usuários e são gravadas amostras para os quatro comandos habilitados (“sobe”, “desce”,

"esquerda" e "direita", além de palavras que devem ser rejeitadas). O usuário pode, a qualquer momento, gravar amostras visando melhorar o desempenho da rede. Há também a possibilidade de se escutar as amostras existentes e apagá-las, evitando que amostras ruins façam parte da base de dados de treinamento do sistema.

3.3.5.2. Treinamento da Rede Neural

O usuário deve solicitar ao sistema que execute o treinamento da rede neural. Nesta etapa o sistema efetivamente é preparado para ser utilizado. O treinamento é realizado aplicando à rede todas as amostras de voz existentes na sua base de dados.

3.3.5.3. Módulo de Teste

Este módulo constitui a demonstração da utilização do sistema em si. Nele o usuário dá comandos de voz e verifica o objeto se movendo na tela de acordo.

Caso o sistema cometa algum erro de reconhecimento, o usuário pode ensinar ao sistema sobre seu próprio erro, clicando em "erro".

Os sinais coloridos à esquerda indicam como o foi o reconhecimento da amostra recém-testada. Vermelho indica que o sistema não a reconheceu, ou seja, que nenhuma saída superou o limite mínimo. Verde significa o oposto, isto é, que o sistema reconheceu. Amarelo significa indecisão, mais de uma saída está em alto.

3.3.5.4. Módulo de Adaptação

Caso o sistema cometa algum erro de reconhecimento, isto é, execute uma ação não correspondente ao comando de voz, o usuário pode "ensinar" a rede, baseado no seu erro, a não cometê-lo novamente.

O que ocorre é que a amostra de voz que resultou em erro de classificação será incorporada à base de dados do sistema, com a semântica que for definida pelo usuário, ou seja, a correta.

Neste módulo o usuário ouve a palavra que está sendo incorporada para ter certeza de que é uma amostra válida e seleciona a semântica correta.

3.4. Testes de Desempenho

Visando melhorar o desempenho do sistema apresentado nas primeiras análises, foi planejado uma bateria de testes de desempenho para avaliar:

- A eficiência do reconhecimento (acertos e erros)

- O comportamento do reconhecimento com a variação de parâmetros fundamentais da rede, como número de neurônios, número de camadas, constante da sigmóide e aprendizado.

O resultado dos testes foi utilizado para obter melhoras de desempenho e para quantificar a eficácia do reconhecimento. Essa quantificação é representada na forma de uma matriz de confusão. A matriz de confusão traz informação tanto sobre o acerto como sobre os erros cometidos no reconhecimento. Nesta matriz é possível encontrar,

por exemplo, a porcentagem de acerto no reconhecimento da palavra “desce”. Também é possível encontrar a porcentagem de vezes que “desce” foi reconhecida como “esquerda”, por exemplo. Desta maneira a eficiência da rede fica bem caracterizada.

Para gerar estes resultados, foi elaborada uma rotina, também em Matlab, que realiza uma série de variações nos parâmetros (número de neurônios, número de camadas, constante da sigmóide e aprendizado) da rede e automaticamente treina a rede com um certo número de amostras e depois executa o teste de reconhecimento. A rotina recebe 5 amostras de 4 palavras: “sobe”, “desce”, “esquerda” e “direita”. Com estas cinco amostras são feitas todas as combinações possíveis de 3 amostras distintas. Com 3 amostras de cada palavra a rotina treina a rede e com as outras duas restantes realiza o teste de reconhecimento. A cada teste são registrados os parâmetros da rede e o resultado do reconhecimento.

Para melhorar ainda mais o resultado desta avaliação, a faixa de variação dos parâmetros foi bem larga, acarretando em extensão do tempo de processamento da bateria de testes. Para cada conjunto de amostras (3 de cada palavra) a rede é treinada para toda faixa de variação dos parâmetros. Isto é interessante pois permitiu a avaliação da influência destes parâmetros no desempenho do reconhecimento.

3.4.1. Matriz de Confusão

A matriz de confusão abaixo condensa o resultado dos testes e permite uma avaliação do desempenho do sistema reconhecedor. A rede foi submetida a 240 amostras, sendo 30 de cada uma das palavras.

resultado / amostra	SOBE	DESCE	ESQUERDA	DIREITA
SOBE	**85%**	0%	15%	0%
DESCE	0%	**100%**	0%	0%
ESQUERDA	0%	0%	**80%**	20%
DIREITA	0%	0%	10%	**90%**

4. Discussão

Diversas alternativas para a tarefa de reconhecimento de voz vêm sendo desenvolvidas. A implementação utilizando redes de neurônios artificiais mostrou ser mais interessante pela analogia com um sistema biológico, pela sua flexibilidade e por que ainda é possível encontrar muitos estudos e desenvolvimentos atuais sobre este tema. A medição de resultados como taxa de acerto, velocidade de reconhecimento e tamanho do vocabulário aceito podem ser usados na comparação entre este método e outros métodos de reconhecimento de voz, porém, este não foi o objetivo deste projeto.

A extração de parâmetros busca enfatizar as características do sinal possibilitando um trabalho mais eficaz do bloco reconhecedor. Além do Modelo de Banco de Filtros

utilizado neste projeto existem outros métodos que desempenham esta função. Durante a etapa de escolha da melhor alternativa, tivemos contato com outras técnicas como por exemplo, extração de parâmetros baseada na FFT. O modelo de extração de parâmetros por banco de filtros foi adotado pelos seus bons resultados, comprovados em outros projetos publicados nesta área, portanto, não houve necessidade de adoção de modelos mais sofisticados.

Os resultados obtidos foram altamente satisfatórios e o sistema mostrou-se robusto. O grau de desenvolvimento atingido permite uma expansão do sistema, atribuindo feições comerciais. Este caráter pode ser conseguido, por exemplo, por uma implementação em hardware com funcionamento em tempo real.

5. Conclusão

O projeto desenvolvido não pode ser resumido apenas ao demonstrativo apresentado. Suas implicações são muito mais abrangentes e tem atuação em diversas áreas. A maior contribuição deste sistema é facilitar a execução de atividades que recebem comandos externos. Dentro desta abordagem enquadram-se o auxílio na navegação em microcomputadores (navegação em menus, internet, alternância entre janelas, execução de comandos padronizados como copiar e colar). Além desta aplicação pode ser implementado o acionamento de equipamentos (máquina de lavar, microondas, TV) e controle de condições ambiente (temperatura, iluminação, ventilação, abertura e fechamento de janelas).

Outra aplicação para o sistema é a identificação de usuários em sistemas de segurança, em que uma palavra é gravada como senha.

A princípio todas as aplicações apresentadas podem parecer simples comodidades agregadas a tarefas comumente desempenhadas sem o uso da voz. No entanto, há campos onde um sistema como este tem caráter essencial. É o caso de sistemas adaptados a deficientes físicos. Estes sistemas podem melhorar a qualidade de vida destas pessoas uma vez que possibilitam acesso a recursos de microinformática, essenciais nos dias de hoje. Além do auxílio de navegação podem ser implementados sistemas que transcrevem a fala do locutor. Isto permite textos sejam redigidos por pacientes com quadros de tetraplegia ou deficiências que impedem interação com o teclado.

Agradecimentos

Ao Professor Emílio Del Moral Hernandez pela orientação e estímulo durante todo o projeto.

Referências Bibliográficas

- S. Haykin, *Neural Networks: a Comprehensive Foundation*

- L. R. Rabiner, R.W. Schafer, *Digital Processing of Speech Recognition*, Englewood Cliffs, Prentice-Hall, 1978

- B. Gold and N. Morgan, *Speech and Audio Signal Processing*, New York, John Wiley & Sons, Inc., 2000

- Flanagan, J.L., *Speech Analysis: Synthesis and Perception* (Springer-Verlag, 1972)

- A. Biem, S. Katagiri: *Filter bank desgn based on discriminative feature extraction.* In *International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 485-489, Apr. 1994.