ALEXANDRE TEIXEIRA MAFRA

**RECONHECIMENTO AUTOMÁTICO DE LOCUTOR
EM MODO INDEPENDENTE DE TEXTO
POR *SELF-ORGANIZING MAPS***

Dissertação apresentada à Escola
Politécnica da Universidade de
São Paulo para obtenção do
Título de Mestre em Engenharia

São Paulo
2002

ALEXANDRE TEIXEIRA MAFRA

**RECONHECIMENTO AUTOMÁTICO DE LOCUTOR
EM MODO INDEPENDENTE DE TEXTO
POR *SELF-ORGANIZING MAPS***

Dissertação apresentada à Escola
Politécnica da Universidade de
São Paulo para obtenção do
Título de Mestre em Engenharia

Área de Concentração:
Engenharia Mecânica

Orientador:
Prof. Dr. Marcelo Godoy Simões

São Paulo
2002

FICHA CATALOGRÁFICA

Mafra, Alexandre Teixeira

Reconhecimento automático de locutor em modo independente de texto por *self-organizing maps* / A.T. Mafra. – São Paulo, 2005.

78 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

**1.Reconhecimento de voz 2.Biometria 3.Redes neurais
I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos II.t.**

À minha família, onde sempre encontrei suporte generoso e confiança, que fundamentam as grandes conquistas pessoais.

A Lúcia, com quem amo viver, pela dedicação e compreensão sem limites.

Aos recém-nascidos Marcos, Bruna e Giovana, que sejam recebidos com generosidade, confiança, dedicação e compreensão, e que os retribuam com suas grandes conquistas.

AGRADECIMENTOS

Agradeço ao professor Dr. Marcelo Godoy Simões pela constante motivação, pelo estímulo à criação e a serenidade na orientação. Sem eles, este trabalho não se tornaria fato.

Ao professor Dr. Jun Okamoto Jr. por sua contribuição, sua incansável disposição e pela seriedade com que vem conduzindo seus alunos da Politécnica há tantos anos.

À professora Dra. Anna Helena Reali Costa e ao professor Dr. Fabio Gagliardi Cozman, por seu envolvimento e suas importantes orientações.

Aos colegas da Uniconsult, companheiros de trabalho e de vida, que me emprestaram suas vozes e sua força. Aos demais amigos, que gentilmente cederam amostras de suas vozes.

RESUMO

Projetar máquinas capazes identificar pessoas é um problema cuja solução encontra uma grande quantidade de aplicações. Implementações em *software* de sistemas baseados em medições de características físicas pessoais (biométricos), estão começando a ser produzidos em escala comercial. Nesta categoria estão os sistemas de Reconhecimento Automático de Locutor, que se usam da voz como característica identificadora.

No presente momento, os métodos mais populares são baseados na extração de coeficientes mel-cepstrais (MFCCs) das locuções, seguidos da identificação do locutor através de *Hidden Markov Models* (HMMs), *Gaussian Mixture Models* (GMMs) ou quantização vetorial. Esta preferência se justifica pela qualidade dos resultados obtidos.

Fazer com que estes sistemas sejam robustos, mantendo sua eficiência em ambientes ruidosos, é uma das grandes questões atuais. Igualmente relevantes são os problemas relativos à degradação de performance em aplicações envolvendo um grande número de locutores, e a possibilidade de fraude baseada em vozes gravadas. Outro ponto importante é embarcar estes sistemas como sub-sistemas de equipamentos já existentes, tornando-os capazes de funcionar de acordo com o seu operador.

Este trabalho expõe os conceitos e algoritmos envolvidos na implementação de um *software* de Reconhecimento Automático de Locutor independente de texto. Inicialmente é tratado o processamento dos sinais de voz e a extração dos atributos essenciais deste sinal para o reconhecimento. Após isto, é descrita a forma pela qual a voz de cada locutor é modelada através de uma rede neural de arquitetura *Self-Organizing Map* (SOM) e o método de comparação entre as respostas dos modelos quando apresentada uma locução de um locutor desconhecido.

Por fim, são apresentados o processo de construção do *corpus* de vozes usado para o treinamento e teste dos modelos, as arquiteturas de redes testadas e os resultados experimentais obtidos numa tarefa de identificação de locutor.

ABSTRACT

The design of machines that can identify people is a problem whose solution has a wide range of applications. Software systems, based on personal physical attributes measurements (biometrics), are in the beginning of commercial scale production. Automatic Speaker Recognition systems fall into this category, using voice as the identifying attribute.

At present, the most popular methods are based on the extraction of mel-frequency cepstral coefficients (MFCCs), followed by speaker identification by Hidden Markov Models (HMMs), Gaussian Mixture Models (GMMs) or vector quantization. This preference is motivated by the quality of the results obtained by the use of these methods.

Making these systems robust, able to keep themselves efficient in noisy environments, is now a major concern. Just as relevant are the problems related to performance degradation in applications with a large number of speakers involved, and the issues related to the possibility of fraud by the use of recorded voices. Another important subject is to embed these systems as sub-systems of existing devices, enabling them to work according to the operator.

This work presents the relevant concepts and algorithms concerning the implementation of a text-independent Automatic Speaker Recognition software system. First, the voice signal processing and the extraction of its essential features for recognition are treated. After this, it is described the way each speaker's voice is represented by a Self-Organizing Map (SOM) neural network, and the comparison method of the models responses when a new utterance from an unknown speaker is presented.

At last, it is described the construction of the speech corpus used for training and testing the models, the neural network architectures tested, and the experimental results obtained in a speaker identification task.

SUMÁRIO

1	Introdução	13
1.1	Reconhecimento Automático de Locutor (RAL)	13
1.2	Classificação do Problema	14
1.2.1	Tipo de Tarefa	14
1.2.2	Modalidades de Texto	15
1.3	Avaliação do Desempenho	15
1.4	Aplicações	15
1.5	Problemas Correlatos	16
1.5.1	Reconhecimento de Voz	16
1.5.2	Entendimento de Voz	17
1.6	Estado da Arte	17
1.7	Visão Geral da Dissertação	18
2	Extração de Atributos dos Sinais de Voz	19
2.1	Digitalizador	20
2.2	Extrator de Atributos	21
2.2.1	Pré-ênfase	24
2.2.2	Janelamento	24
2.2.3	DFT	27
2.2.4	Bancos Mel-Frequency	29
2.2.5	$\log \cdot $	32
2.2.6	DCT e MFCCs	33
	Delta Cepstrum	34
3	Reconhecimento de Locutor por <i>Self-Organizing Maps</i>	36
3.1	Visão Geral	36
3.2	Redes Neurais Artificiais	36
3.2.1	Unidades de Processamento	37
	Funções de Ativação	38

3.2.2	Arquiteturas de Redes	39
	Camadas	39
	Redes <i>Feedforward</i> de Uma Camada	40
	Redes <i>Feedforward</i> Multi-camada	40
	Redes Recorrentes	41
3.2.3	Processos de Aprendizado de Redes Neurais	41
	Aprendizado Supervisionado	42
	Aprendizado Não Supervisionado	42
	Aprendizado Competitivo	43
3.3	Redes Neurais para Reconhecimento de Locutor	44
3.3.1	<i>Multi-Layer Perceptron</i> (MLP)	44
3.3.2	<i>Self-Organizing Map</i> (SOM) e <i>Learning Vector Quantization</i> (LVQ)	44
3.3.3	<i>Time-Delay Neural Network</i> (TDNN)	45
3.3.4	<i>Recurrent Neural Network</i> (RNN)	46
3.3.5	<i>Neural Tree Network</i> (NTN)	46
3.4	Quantização Vetorial (<i>Vector Quantization</i> – VQ)	47
3.5	A Arquitetura <i>Self-Organizing Map</i> (SOM)	47
3.5.1	Algoritmo de Treinamento	48
	Competição	48
	Cooperação	49
	Adaptação	50
3.5.2	Propriedades do Mapa de Atributos	51
3.6	<i>Self-Organizing Maps</i> para Reconhecimento Automático de Locutor . .	52
3.6.1	Princípio de Funcionamento	52
3.6.2	Normalização	53
3.6.3	Treinamento	53
3.6.4	Teste	54
4	Procedimento Experimental	55
4.1	<i>Corpus</i> de Amostras de Voz	55
4.1.1	Condições do Ambiente	55
4.1.2	Composição do Conjunto de Locutores	55
4.1.3	Frases Usadas	56
4.2	<i>Software</i> e <i>Hardware</i> Usados	58

4.3	Digitalização e Extração dos MFCCs	58
4.4	Arquiteturas de SOMs Usadas	59
4.5	Treinamento	59
4.5.1	Análise dos Resultados do Treinamento	61
4.6	Resultados dos Testes	61
4.6.1	Análise dos Resultados dos Testes	61
5	Conclusões	69
5.1	Conclusões Gerais	69
5.2	Sobre o uso de uma SOM por locutor	70
5.3	Direções Futuras para Pesquisa	71

LISTA DE FIGURAS

1	Diagrama de blocos do Digitalizador	20
2	Palavra “zero”, digitalizada a $22.05kHz$ com resolução de 8 bits	22
3	Espectro de potência da palavra “zero”	22
4	Diagrama de blocos do Extrator de Atributos	23
5	Palavra “zero” após pré-ênfase	25
6	Espectro de potência da palavra “zero” após pré-ênfase	25
7	Funções de Janelamento	28
8	Segmento da palavra “zero” sem janelamento	28
9	Segmento da palavra “zero” após janelamento	30
10	Espectro de potência de segmento da palavra “zero”	30
11	Filtros de bandas críticas da escala Mel	31
12	Potência dos filtros de bandas críticas de segmento da palavra “zero”	31
13	Log da potência dos filtros de bandas críticas de segmento da palavra “zero”	32
14	Coefficientes mel-cepstrais de segmento da palavra “zero”	33
15	Coefficientes mel-cepstrais da palavra “zero”	35
16	Modelo não linear de um neurônio	38
17	Rede neural <i>feedforward</i> de uma camada	40
18	Rede neural <i>feedforward</i> multi-camada	41
19	Rede neural TDNN com entrada unidimensional	45
20	<i>Recurrent Neural Network</i> (RNN)	46
21	Arquitetura de uma <i>Self-Organizing Map</i>	48
22	Retícula bidimensional de uma <i>Self-Organizing Map</i>	50
23	Evolução do erro de quantização por segmento para cada arquitetura de SOM	60
24	Média do erro de quantização após treinamento em função do número de unidades da arquitetura	65
25	Taxa de erro de identificação em função da duração do conjunto de treinamento	66

26	Taxa de erro de identificação em função do número de unidades da arquitetura usada, para cada conjunto de treinamento	66
27	Taxa de erro de identificação em função da duração média da locução .	68

LISTA DE TABELAS

1	Composição do conjunto de locutores do <i>corpus</i> de amostras de voz . . .	56
2	Conjunto VAR1 de frases variáveis por locutor	57
3	Conjunto BAL1, foneticamente balanceado, de frases comuns a todos os locutores	57
4	Conjunto BAL2, foneticamente balanceado, de frases comuns a todos os locutores	57
5	Conjunto BAL3, foneticamente balanceado, de frases comuns a todos os locutores	58
6	Características dos conjuntos que compõem o <i>corpus</i>	59
7	Arquiteturas de SOM utilizadas e seus parâmetros de treinamento . . .	60
8	Resultados do treinamento da arquitetura SOM 16 sobre os conjuntos VAR1, BAL1 e BAL2, para cada um dos locutores	62
9	Resultados do treinamento da arquitetura SOM 25 sobre os conjuntos VAR1, BAL1 e BAL2, para cada um dos locutores	62
10	Resultados do treinamento da arquitetura SOM 36 sobre os conjuntos VAR1, BAL1 e BAL2, para cada um dos locutores	63
11	Resultados do treinamento da arquitetura SOM 64 sobre os conjuntos VAR1, BAL1 e BAL2, para cada um dos locutores	63
12	Porcentagens de identificação correta do locutor para a SOM 16	64
13	Porcentagens de identificação correta do locutor para a SOM 25	64
14	Porcentagens de identificação correta do locutor para a SOM 36	64
15	Porcentagens de identificação correta do locutor para a SOM 64	64
16	Taxa de erro de identificação em função da duração média da locução .	68

LISTA DE ABREVIATURAS

ANN	– Artificial Neural Network
ASR	– Automatic Speech Recognition
BALn	– Conjunto foneticamente balanceado de frases comuns
DCT	– Discrete Cosine Transform
DFT	– Discrete Fourier Transform
FFT	– Fast Fourier Transform
GMM	– Gaussian Mixture Model
HMM	– Hidden Markov Model
IDFT	– Inverse Discrete Fourier Transform
LVQ	– Learning Vector Quantization
MFCC	– Mel-Frequency Cepstral Coefficient
MLP	– Multi-Layer Perceptron
NTN	– Neural Tree Network
PU	– Processing Unit
RAL	– Reconhecimento Automático de Locutor
RNA	– Rede Neural Artificial
RNN	– Recurrent Neural Network
SOM	– Self-Organizing Map
stRC	– short term Real Cepstrum
TDNN	– Time-Delay Neural Network
VAR1	– Conjunto de frases variáveis
VQ	– Vector Quantization

1 INTRODUÇÃO

A identificação do operador é um problema comum a muitos projetos de Engenharia. Uma grande variedade de exemplos de métodos de identificação pessoal é facilmente encontrada no cotidiano: documentos, chaves, cartões magnéticos, códigos de identificação e senhas de acesso.

Todos estes métodos possuem uma característica em comum: a identificação da pessoa está associada a alguma coisa que ela *possua* ou *saiba*, obrigando-a a portar objetos e decorar códigos. Diversos problemas decorrem do esquecimento ou confusão destes códigos ou da perda, cópia ou fraude dos objetos de identificação.

Neste contexto surgem as soluções baseadas em métodos biométricos (PHILLIPS, 2000), ou seja, baseados na medição e reconhecimento de características físicas individuais como forma de identificação. Reconhecedores de impressões digitais, de íris, de face e de locutor se encaixam nesta categoria. É também comum o uso combinado destes métodos, para o aumento do nível de segurança.

Dentre os métodos biométricos de identificação, a voz recebe bastante atenção dos pesquisadores da área. Esta atenção se deve principalmente à facilidade com que é produzida, capturada, processada e transmitida no estado tecnológico atual.

1.1 Reconhecimento Automático de Locutor (RAL)

Os sistemas de Reconhecimento Automático de Locutor (RAL) têm como objetivo a determinação automática do indivíduo emissor de uma determinada locução, materializada num sinal de voz, através da comparação entre características extraídas da locução atual e locuções anteriores.

Desta forma, uma grande área inicial de pesquisa consiste na modelagem (BORDEN; HARRIS, 1980; DENES; PINSON, 1993), análise (FLANAGAN, 1972; OPPENHEIM; SCHAFER, 1998) e técnicas de redução do ruído incorporado nestes sinais de voz (DELLER JR.; HANSEN; PROAKIS, 1993, cap. 8). Outra grande frente se dedica

à definição das características mais representativas destes sinais (RABINER; JUANG, 1993, cap. 3 e 4), objetivando ressaltar as peculiaridades e reduzir as redundâncias, simplificando a tarefa de classificação do locutor.

Para efetuar o reconhecimento em si, os principais métodos usados são baseados em *Hidden Markov Models* (HMMs) (RABINER, 1989), *Gaussian Mixture Models* (GMMs) (REYNOLDS; HECK, 2000) e *Redes Neurais Artificiais* (RNAs) (FARRELL, 2000).

1.2 Classificação do Problema

Segue a classificação do problema, apresentada em Furui (1996) e amplamente aceita em outras referências no assunto.

1.2.1 Tipo de Tarefa

De maneira geral, um sistema de RAL pode ser aplicado para a *identificação* ou para a *verificação* de um locutor.

Identificação de locutor: nesta categoria, é assumido um conjunto de locutores, cada qual com um conjunto de padrões que o representam, e a tarefa do sistema consiste em comparar a locução proposta contra estes padrões, decidindo a qual locutor pertence o padrão mais semelhante. Existem duas variações desta tarefa: a identificação em *conjunto fechado*, em que se supõe que somente locutores pertencentes ao conjunto alvo participam tanto do aprendizado quanto da identificação, e a identificação em *conjunto aberto*, em que um elemento adicional é acrescentado, representando qualquer locutor que não pertença ao conjunto de locutores alvo.

Verificação de locutor: no caso da verificação ou autenticação, o sistema é responsável por comparar uma locução de entrada com o padrão associado à identidade proposta para esta locução, chegando a uma decisão binária do tipo aceito x rejeitado dentro de uma margem de segurança definida (GISH; SCHMIDT, 1994). Em resumo o sistema deve confirmar se a locução proposta realmente pertence ao locutor proposto ou negar esta associação, atribuindo a locução a um impostor.

1.2.2 Modalidades de Texto

Uma segunda classificação dos sistemas RAL divide-os em *dependentes* ou *independentes* conteúdo textual da emissão.

Dependentes de texto: o locutor deve pronunciar um determinado texto, para que o reconhecimento ocorra. Este texto deve ser fixo ou proposto pelo sistema. O sistema já deve ter sido treinado com o texto específico, dito pelo locutor.

Independentes de texto: o locutor deve ser reconhecido independentemente do texto que esteja falando. O sistema não leva em consideração o texto pronunciado.

1.3 Avaliação do Desempenho

Para avaliar o desempenho de um sistema RAL, devem ser considerados os seguintes aspectos:

Qualidade do sinal de voz: características do microfone e do canal de transmissão, tipo e nível de ruído ambiente e a variação entre os sinais de treinamento e teste.

Modalidade de texto: dependente ou independente de texto.

Duração das locuções: duração e número de sessões de treinamento e verificação.

População de locutores: número de locutores e composição do conjunto segundo sexo, idade e origem, entre outros.

1.4 Aplicações

Autenticação de transações comerciais: sistemas de identificação automática de locutor podem ser usados para autenticar o locutor em chamadas telefônicas de longa distância, validar transações comerciais com pagamento através de cartão de crédito por telefone e validar aplicações financeiras via telefone.

Controle de acesso: sistemas de identificação e verificação como meio de acesso para ambientes físicos ou redes de computadores. Em hospitais, podem ser usados para dar acesso ao registro de informações de pacientes ou para habilitar a prescrição de medicamentos e exames, restritas ao pessoal autorizado.

Monitoramento: sistemas de identificação usados no monitoramento de *call-centers*, registrando o atendente e o tempo gasto nas chamadas. Ainda como monitoramento, podem ser usados na verificação do cumprimento de prisão condicional doméstica, ou no monitoramento das chamadas feitas dentro de uma prisão, rastreando e identificando automaticamente os contatos dos prisioneiros.

Fonte de informações: sistemas de identificação usados para determinar os trechos de cada locutor em arquivos contendo gravações de conversas entre vários locutores.

Aplicação forense: sistemas de verificação são usados atualmente para comparar trechos de gravações telefônicas com amostras fornecidas pelos supostos locutores, confirmando ou negando a identidade dentro de um certo grau de aceitação.

Aplicações híbridas: sistemas RAL, associados a sistemas de reconhecimento de voz, podem fornecer um alto grau de segurança em tarefas de verificação em contextos dependentes de texto, pela verificação simultânea do locutor e do conhecimento de palavras secretas, interpretadas pelo reconhecedor de voz.

1.5 Problemas Correlatos

1.5.1 Reconhecimento de Voz

O objetivo da tecnologia de reconhecimento de voz, segundo Deller Jr., Hansen e Proakis (1993, p. 601), é o de “criar máquinas que possam receber informações faladas e agir de forma apropriada de acordo com estas informações”. Os sistemas de Reconhecimento de Voz (Automatic Speech Recognition, ASR) (RABINER; JUANG, 1993; DELLER JR.; HANSEN; PROAKIS, 1993; LEE; SOONG; PALIWAL, 1996; DE MORI, 1998) tentam fazer com que um computador reconheça, com 100% de acuracidade, todas as palavras que possam ser entendidas por qualquer pessoa, independentemente do tamanho do vocabulário, ruídos presentes, características ou sotaque do locutor, e preferencialmente em tempo real.

Estes têm sido alvo de intensa pesquisa na última década, motivada pela grande quantidade de possíveis aplicações comerciais desta tecnologia (SRINIVASAN; BROWN, 2002). A concepção e implementação deste tipo de *software* está em fase de amadurecimento, mas o problema do reconhecimento ainda é considerado bastante aberto e sujeito a novas abordagens para a sua solução. Ainda assim, aplicações comerciais estão recebendo grandes investimentos e experimentando rápida popularização,

principalmente aquelas embarcadas em *handhelds* (KUMAGAI, 2002).

1.5.2 Entendimento de Voz

Considerando a riqueza dos aspectos cognitivos da comunicação humana falada, o reconhecimento de voz como mera transcrição das palavras parece bastante limitado, pois ignora o real significado e a intenção de comunicar do locutor. A área de pesquisa de *Entendimento de Voz* busca representar as intenções da comunicação de uma forma mais rica do que a simples transcrição do sinal de voz. Obviamente, esta tarefa é extremamente ambiciosa, dado que os processos do pensamento humano e intenções de comunicação são muito pouco compreendidos.

1.6 Estado da Arte

Atualmente, o método mais eficaz para o reconhecimento de locutor depende principalmente da modalidade de texto associada ao problema.

Os *Hidden Markov Models* (HMMs) demonstram os melhores resultados em aplicações dependentes de texto. Os HMMs são modelos estatísticos, com grande capacidade de modelagem das dependências temporais associadas aos sinais de voz. Para uma introdução sobre a aplicação e os resultados do uso de HMMs para reconhecimento automático de locutor veja Naik, Netch e Doddington (1989), Rosenberg, Lee e Soong (1990), Rosenberg, Lee e Gokeen (1991) e Matsui e Furui (1994).

Os *Gaussian Mixture Models* (GMMs) são também modelos estatísticos, em que as probabilidades de ocorrência dos vetores de atributos para cada locutor são modeladas como combinações ponderadas de variáveis aleatórias vetoriais com pdfs Gaussianas. Usados com excelentes resultados em aplicações independentes de texto, conforme demonstrado por Bimbot, Magrin-Chagnolleau e Mathan (1995), Reynolds (1995) e Reynolds, Quatieri e Dunn (2000).

As Redes Neurais Artificiais (RNAs) são modelos conexionistas não lineares, com grande capacidade de reconhecimento e classificação de padrões estáticos. Muitas arquiteturas de RNAs foram experimentadas em reconhecimento de locutor, sendo que os melhores resultados são conseguidos pelo uso de arquiteturas baseadas em Quantização Vetorial para aplicações independentes de texto. Uma das primeiras referências sobre o reconhecimento automático de locutor através de Quantização Vetorial pode ser vista em Soong (1985). Bennani e Gallinari (1990), Anderson e Patterson (1994) e

Lapidot, Guterman e Cohen (2002) apresentam os resultados do uso de RNAs executando reconhecimento automático de locutor através de Quantização Vetorial. Segundo Homayounpour e Chollet (1995), seu desempenho é comparável ao dos GMMs.

Um resumo sobre o estado da arte em RAL, parte de uma coleção de artigos sobre o estado da arte em tecnologias relacionadas à linguagem humana, pode ser encontrada em Furui (1996). Outras referências sobre os métodos usados incluem Furui (1994), Campbell Jr. (1997), Furui (1997) e Farrell (2000).

1.7 Visão Geral da Dissertação

A dissertação está estruturada da seguinte forma:

Capítulo 2: descreve o processamento dos sinais de voz, focando na extração dos atributos relevantes para reconhecimento automático de locutor.

Capítulo 3: faz uma breve introdução às redes neurais, citando as principais arquiteturas usadas em reconhecimento automático de locutor e aprofundando-se na arquitetura *Self-Organizing Map*. Apresenta então a modelagem das vozes dos locutores através de redes neurais de arquitetura SOM e o critério de comparação entre as respostas dos modelos dos locutores quando apresentados a uma locução de locutor desconhecido.

Capítulo 4: relata a forma como foi elaborado o *corpus* de amostras de vozes dos locutores usadas para treinamento e teste, as arquiteturas usadas, o procedimento experimental conduzido sobre este conjunto e os resultados obtidos na tarefa de identificação de locutor em conjunto fechado.

Capítulo 5: apresenta as conclusões sobre os resultados dos experimentos e direções futuras para pesquisa.

2 EXTRAÇÃO DE ATRIBUTOS DOS SINAIS DE VOZ

Para que sejam usados para reconhecimento de locutor, os sinais de voz devem ser digitalizados e pré-processados. O objetivo do pré-processamento é obter uma representação paramétrica dos sinais, que reduza redundâncias, mantendo informações estatísticas suficientes para o reconhecimento. Do ponto de vista do reconhecedor, estas representações são os atributos do sinal de voz (*features*), que constituem o objeto de reconhecimento.

Um bom conjunto de atributos (REYNOLDS; HECK, 2000) deve ser prático, robusto e seguro, o que significa que estas características devem ocorrer naturalmente e freqüentemente no sinal de voz, devem ser facilmente mensuráveis, não devem ser muito afetadas pela idade ou pela saúde do locutor, devem ser pouco afetadas pelo ruído de fundo ou pelo ruído introduzido pelo canal de comunicação e não devem ser facilmente imitáveis.

Diversas representações paramétricas já foram experimentadas em sistemas de reconhecimento de voz e locutor, sendo que a que apresenta os melhores resultados na maioria dos casos são os *Coeficientes Mel-Cepstrais* (*Mel-Frequency Cepstral Coefficients* – MFCCs) e suas derivadas. Mermelstein e Davis (1980) e Picone (1993) apresentam uma revisão sobre estas representações e justificam a escolha dos MFCCs. Bechetti e Ricotti (1999) apresentam uma síntese prática do processo de extração de MFCCs, enquanto Deller Jr., Hansen e Proakis (1993) trazem uma demonstração mais formal deste mesmo processo.

A extração de atributos de um sistema de reconhecimento de voz é executada por 2 blocos:

- Digitalizador
- Extrator de Atributos (*Feature Extractor*)

2.1 Digitalizador

Responsável pela captura e digitalização (discretização do tempo e da amplitude) do sinal de voz (figura 1), normalmente integrado numa placa de som (com exceção do Banco de Polifones). Uma vez amostrado e discretizado, o sinal de voz pode ser processado digitalmente.

Figura 1: Diagrama de blocos do Digitalizador

Microfone. Recebe os sinais enviados pelo locutor através de ondas de pressão do ar $p(t)$ e os converte em sinais analógicos de tensão elétrica $x_c(t)$, onde t é o índice de tempo contínuo.

Pré-amplificador. Filtro analógico de ganho positivo na entrada de $x_c(t)$.

Anti-aliasing. Filtro analógico (Butterworth, Chebyshev, Inverse Chebyshev, Cauer, Bessel-Thomson) que corta frequências altas (acima da largura de banda relevante), evitando que estas frequências sejam rebatidas para o espectro relevante durante a análise espectral, introduzindo um tipo de ruído conhecido como *alias* (TEXAS INSTRUMENTS INC., 1995).

Sampler/Holder. Amostra (*sampler*) o sinal $x_c(t)$ em intervalos T_c , com frequência de amostragem $f_c = 1/T_c$. O instante de amostragem t é discretizado, sendo dado por $t = nT_c$, onde n é o índice de tempo discreto da amostra (também chamado de tempo *normalizado*). O sinal é mantido estável (*holder*) durante o intervalo necessário para a conversão A/D. O sinal passa a ser representado por $x_c(n)$. São comuns taxas de amostragem de $8kHz$, $11.025kHz$, $12kHz$, $16kHz$ e $22.05kHz$. Mais usualmente, durante procedimentos experimentais, as amostras são tomadas a $22.05kHz$ e então têm sua frequência de amostragem baixada artificialmente para a frequência desejada através de *downsampling*.

Conversor A/D. Recebe os sinais amostrados $x_c(n)$ e os quantiza com uma determinada resolução, gerando o sinal $x(n)$. Resoluções de 8, 12 e 16 bits são comuns.

Banco de Polifones. Uma vez digitalizados, os sinais podem ser armazenados em memória permanente (*hard drive*), em arquivos (.wav, .mp3), guardando referência ao locutor e à locução (figura 2).

2.2 Extrator de Atributos

Cadeia de processamento (figura 4) que recebe os sinais de voz digitalizados $x(n)$, executa a segmentação e extrai os vetores de coeficientes mel-cepstrais (MFCCs) $\mathbf{c}_w$ dos segmentos, que são os atributos a serem reconhecidos.

Seja o conjunto $\mathbb{X}$ de locuções $\mathbf{X}$ de diversas palavras ou frases p ($1 \leq p \leq P$), emitidas por diferentes locutores l ($1 \leq l \leq L$), por diversas vezes i ($1 \leq i \leq I$).

$$\mathbb{X} = \{\mathbf{X}_{l,p,i}\}, 1 \leq l \leq L, 1 \leq p \leq P, 1 \leq i \leq I \quad (2.1a)$$

$$\mathbf{X} = \mathbf{X}_{l,p,i} = \{x_{l,p,i}(n)\}, 1 \leq n \leq N_{l,p,i} = \quad (2.1b)$$

$$= \{x_{l,p,i}(1), \dots, x_{l,p,i}(n), \dots, x_{l,p,i}(N_{l,p,i})\} \quad (2.1c)$$

Figura 2: Palavra “zero”, digitalizada a 22.05kHz com resolução de 8 bits

Figura 3: Espectro de potência da palavra “zero”

Figura 4: Diagrama de blocos do Extrator de Atributos

Exemplificando, $\mathbf{X}_{4,2,3}$, representa o conjunto de $N_{4,2,3}$ amostras do sinal $x_{4,2,3}(n)$ que compõem a locução $i = 3$ da palavra ou frase $p = 2$ pelo locutor $l = 4$. Para simplicidade de notação, será usado $x(n)$ significando $x_{l,p,i}(n)$ e N significando $N_{l,p,i}$ deste ponto em diante.

2.2.1 Pré-ênfase

Observa-se que, para sinais de voz, a energia carregada pelas altas frequências é pequena quando comparada com as baixas frequências. A pré-ênfase das frequências altas é necessária para que se obtenha amplitudes mais homogêneas das frequências formantes, pois informações importantes sobre a locução também estão presentes nas altas frequências. Isto pode ser feito através de um filtro digital (OPPENHEIM; SCHAFER, 1998), cuja função de transferência no domínio z é

$$H(z) = 1 - a \cdot z^{-1}, 0 \leq a \leq 1 \quad (2.2)$$

sendo a o parâmetro responsável pela pré-ênfase, da ordem de 0.95.

No domínio do tempo, o filtro é implementado através de

$$\mathbf{X}' = \{x'(n)\}, 1 \leq n \leq N \quad (2.3a)$$

$$x'(n) = x(n) - ax(n-1) \quad (2.3b)$$

com $x(0) = 0$.

O efeito da pré-ênfase pode ser observado comparando-se o sinal original (figura 2) e o sinal após a pré-ênfase (figura 5) e os respectivos espectros de potência (figuras 3 e 6).

2.2.2 Janelamento

Após a pré ênfase, as locuções $\mathbf{X}'$ são divididas em segmentos (*frames*) para que se possa fazer a análise *short term* do sinal. Esta segmentação é necessária pois o espectro médio do sinal ao longo de toda a observação (*long term*) encobre variações espectrais que ocorrem durante curtos intervalos de tempo e que são de grande relevância para o reconhecimento. Dentro de uma janela suficientemente pequena (aproximadamente 20ms), as variações no espectro de um sinal de voz podem ser consideradas desprezíveis, tornando válida a análise de Fourier. São comuns janelas entre 10ms e 30ms.

Figura 5: Palavra “zero” após pré-ênfase

Figura 6: Espectro de potência da palavra “zero” após pré-ênfase

Tomando-se segmentos $\mathbf{Y}_w$, de largura $N_w < N$, com N_w sendo uma potência de 2, sobrepostos segundo uma razão s , com $0 \leq s < 1$,

$$s = \frac{\text{número de amostras sobrepostas}}{N_w} \quad (2.4)$$

cada locução $\mathbf{X}'$ é segmentada em $W = W_{l,p,i}$ janelas, sendo

$$W = \frac{N}{(1-s)N_w} \quad (2.5)$$

Os valores de s mais utilizados são 0.25, 0.50 e 0.75.

O índice de tempo discreto dentro do segmento n' guarda a seguinte relação com o índice de tempo discreto da locução:

$$n' = n - (w-1)(1-s)N_w, 1 \leq n' \leq N_w \quad (2.6)$$

Desta forma, a locução completa original é representada por um conjunto $\mathbb{Y}$ de segmentos de locução $\mathbf{Y}_w$ (figura 8)

$$\mathbb{Y} = \{\mathbf{Y}_w\}, 1 \leq w \leq W \quad (2.7a)$$

$$\mathbf{Y}_w = \{y_w(n')\}, 1 \leq n' \leq N_w = \quad (2.7b)$$

$$= \{y_w(1), \dots, y_w(n'), \dots, y_w(N_w)\} \quad (2.7c)$$

Para evitar o efeito do fenômeno de *leakage* (introdução de ruídos em frequência devidos à segmentação), utilizam-se funções de janelamento $\zeta(n')$ (OPPENHEIM; SCHAFER, 1998). Os valores da amostra dentro dos segmentos são multiplicados ponto a ponto pela resposta impulsiva no tempo da função de janelamento (figura 9)

$$y_w(n') = \zeta(n')x'(n') \quad (2.8)$$

As funções de janelamento seguintes (figura 7) já foram aplicadas em reconhecimento automático de locutor, sendo que as de uso mais freqüente são a janela de *Hamming* e a janela de *Hanning*:

- Janela de Bartlett

$$\zeta_{Bartlett}(n') = \begin{cases} 2n'/N_w, & n' < N_w/2 \\ 2 - 2n'/N_w, & n' \geq N_w/2 \end{cases} \quad (2.9)$$

- Janela de Hamming

$$\zeta_{Hamming}(n') = 0.54 - 0.46 \cos \left[\frac{2\pi}{N_w} \cdot \left(n' - \frac{N_w}{2} \right) \right] \quad (2.10)$$

- Janela de Hanning

$$\zeta_{Hanning}(n') = 0.5 - 0.5 \cos \left[\frac{2\pi}{N_w} \cdot \left(n' - \frac{N_w}{2} \right) \right] \quad (2.11)$$

- Janela de Blackman

$$\begin{aligned} \zeta_{Blackman}(n') = 0.42 - 0.5 \cos \left[\frac{2\pi}{N_w} \cdot \left(n' - \frac{N_w}{2} \right) \right] + \\ + 0.08 \cos \left[\frac{4\pi}{N_w} \cdot \left(n' - \frac{N_w}{2} \right) \right] \end{aligned} \quad (2.12)$$

2.2.3 DFT

Tendo agora um conjunto de janelas de amostras do sinal $\mathbb{Y} = \{\mathbf{Y}_w\}$, procede-se a análise de Fourier de cada uma delas. É calculada a DFT (*Discrete Fourier Transform*) de cada segmento, através de seu algoritmo rápido: a FFT (COOLEY; TUKEY, 1965). Para maiores detalhes, vide Oppenheim e Schafer (1998, capítulo 8), Deller Jr., Hansen e Proakis (1993, seção 1.1.4) e Press (1992, capítulo 12).

As amostras no domínio do tempo n' são levadas para o domínio da frequência pela transformação

$$z_w(\omega) = \sum_{n'=1}^{n'=N_w} y_w(n') e^{-j(2\pi/N_w)(n'-1)\omega} \quad (2.13)$$

onde os coeficientes $z_w(\omega)$ são números complexos.

A aplicação da DFT gera, para cada locução, um conjunto $\mathbb{Z}$ de janelas $\mathbf{Z}_w$ (figura 10) contendo $N_w/2$ componentes do espectro de potência do sinal (somente a metade dos componentes é utilizada, pois o sinal de entrada é composto por números reais,

Figura 7: Funções de Janelamento, 512 pontos

Figura 8: Segmento 35 da palavra “zero” (Y_{35}), sem janelamento

tendo o espectro simétrico em relação à frequência de Nyquist $f_c/2$),

$$\mathbb{Z} = \{\mathbf{Z}_w\}, 1 \leq w \leq W \quad (2.14a)$$

$$\mathbf{Z}_w = \{|z_w(\omega)|^2\}, 1 \leq \omega \leq \Omega \quad (2.14b)$$

com $\Omega = \pi f_c$ e com o intervalo entre frequências consecutivas dado pela relação

$$\Delta\omega = \frac{2\pi f_c}{N_w} \quad (2.15)$$

e $\omega = n'\Delta\omega$.

2.2.4 Bancos Mel-Frequency

No estudo da dinâmica do sistema auditivo humano, definiu-se uma escala psicoacústica de sensibilidade do ouvido para diversas frequências do espectro audível, conhecida como escala Mel (STEVENS; VOLKMAN, 1940; KOENIG, 1949). Um *mel* é uma unidade de medida de frequência percebida para uma determinada frequência de entrada (recebida). A interpolação dos resultados obtidos por este estudo leva aproximadamente à seguinte relação entre a frequência recebida (f_{Hz}) e a frequência percebida (f_{mel}):

$$f_{mel} = 1000 \frac{\ln\left(1 + \frac{f_{Hz}}{700}\right)}{\ln\left(1 + \frac{1000}{700}\right)} \quad (2.16)$$

Foi demonstrado (SHIKANO, 1985) que o desempenho de sistemas de reconhecimento de voz aumenta com o uso da escala Mel, aliada ao uso de bancos de filtros com uma função de envelope triangular. Estes filtros fazem a soma dos coeficientes, ponderados pela função triangular centrada na frequência principal, com valor 1 neste ponto.

Os filtros têm suas frequências centrais espaçadas de $100mel$ ou mais, cobrindo o espectro entre $200Hz$ e $7kHz$, onde a maior quantidade de energia se concentra. Evidentemente, a frequência superior é limitada pela frequência de Nyquist (metade da frequência de amostragem). Assim, são comumente encontradas aplicações onde a frequência superior varia de $3kHz$ até $11kHz$.

Figura 9: Segmento 35 da palavra “zero” (Y_{35}), após janela de Hamming. Observar a atenuação nas extremidades do segmento, diminuindo a potência do ruído introduzido pela segmentação.

Figura 10: Espectro de Potência do segmento 35 da palavra “zero”, após janela de Hamming

Figura 11: Filtros de bandas críticas da escala Mel

Figura 12: Potência total nos filtros de bandas críticas do segmento 35 da palavra "zero"

2.2.5 $\text{Log}|\cdot|$

Para o cálculo do *cepstrum* real (*short term Real Cepstrum* – stRC) aplica-se a função log sobre o espectro de potência (figura 10) das janelas $\mathbf{Z}_w$. Os sinais de voz são formados pela convolução no tempo entre o sinal de excitação produzido pela traquéia e a resposta impulsiva instantânea do trato vocal. Estes sinais, convoluídos no domínio do tempo, aparecem multiplicados no domínio da frequência. A aplicação do log sobre o espectro transforma esta multiplicação em soma, permitindo a separação linear dos componentes do sinal de excitação e do sinal de modulação do trato vocal. A saída desta operação é, para cada locução, um conjunto $\mathbb{Z}'$ de janelas $\mathbf{Z}'_w$ (figura 13) contendo os coeficientes transformados

$$\mathbb{Z}' = \{\mathbf{Z}'_w\}, 1 \leq w \leq W \quad (2.17a)$$

$$\mathbf{Z}'_w = \{z'_w(\omega)\}, 1 \leq \omega \leq \Omega \quad (2.17b)$$

$$z'_w(\omega) = \log(|z_w(\omega)|^2) = 2 \log(|z_w(\omega)|) \quad (2.17c)$$

O coeficiente 2 da equação anterior pode ser desprezado por não ser significativo em termos de reconhecimento e por sua eliminação reduzir a complexidade computacional associada à extração da raiz quadrada para o cálculo da magnitude.

Figura 13: Log da potência total nos filtros de bandas críticas do segmento 35 da palavra “zero”

2.2.6 DCT e MFCCs

Sobre os logaritmos das energias dos filtros de bandas críticas $z'_w(\omega)$ é calculada a seguir a transformada inversa de Fourier IDFT (*Inverse Discrete Fourier Transform*), gerando os coeficientes mel-cepstrais (MFCCs). Como as janelas geradas pela operação anterior são compostas apenas por valores reais, a IDFT se reduz à DCT (*Discrete Cosine Transform*).

O espectro do logaritmo do trato vocal possui uma variação suave e de baixa frequência, enquanto o da excitação é altamente variável e quase periódico, principalmente para as vogais. Assim, a resposta do trato vocal pode ser obtida pela simples retenção dos primeiros M coeficientes mel-cepstrais. Esta operação gera um conjunto $\mathbb{C}$ (figura 15) de vetores $\mathbf{c}_w$ (figura 14) que são os vetores de coeficientes mel-cepstrais.

$$\mathbb{C} = \{\mathbf{c}_w\}, 1 \leq w \leq W \quad (2.18a)$$

$$\mathbf{c}_w = [c_{w,m}]_{m=1}^M = \quad (2.18b)$$

$$= [c_{w,1}, \dots, c_{w,m}, \dots, c_{w,M}]^T \quad (2.18c)$$

Figura 14: Coeficientes mel-cepstrais do segmento 35 da palavra “zero” ($\mathbf{c}_{35}$)

O número M de coeficientes é escolhido de acordo com a distância em frequência entre os filtros do banco, sendo muito usados $M = 8$, $M = 12$, $M = 14$ e $M = 16$

coeficientes mel-cepstrais.

Delta Cepstrum

Além dos coeficientes mel-cepstrais em si, os sistemas atuais usam estimativas de suas derivadas temporais como componentes adicionais do vetor de atributos, colocando-as num patamar de igual relevância ao dos coeficientes em si (JUANG; RABINER; WILPON, 1987; RABINER; WILPON; SOONG, 1989; BOCCHIERI; WILPON, 1993). O objetivo desta inclusão é acrescentar alguma informação temporal sobre a variação dos MFCCs.

Estas componentes, conhecidas como delta *cepstrum* ou delta-MFCCs são facilmente computáveis através da variação dos coeficientes entre o segmento atual e um ou dois segmentos precedentes e consequentes.

Seja δw a distância entre o segmento atual w e os segmentos precedente e subsequente considerados para a diferenciação. Então, o delta-MFCC da componente m no segmento w é definido por

$$c'_{w,m} = c_{w+\delta w,m} - c_{w-\delta w,m} \quad (2.19)$$

Supondo-se $\delta w = 1$, para os casos particulares em que $w = 1$ e $w = W$, define-se

$$c'_{1,m} = \frac{c_{2,m} - c_{1,m}}{2} \quad (2.20a)$$

$$c'_{W,m} = \frac{c_{W,m} - c_{W-1,m}}{2} \quad (2.20b)$$

Desta forma, a dimensão dos vetores de MFCCs do conjunto $\mathbb{C}_w$ é dobrada ($2M$), e o conjunto é então redefinido como

$$c_{w,m} = \begin{cases} c_{w,m}, & \text{para } 1 \leq m \leq M \\ c'_{w,m}, & \text{para } M+1 \leq m \leq 2M \end{cases} \quad (2.21a)$$

$$\mathbf{c}_w = [c_{w,m}]_{m=1}^{2M} = [c_{w,1}, \dots, c_{w,M}, c'_{w,1}, \dots, c'_{w,M}]^T \quad (2.21b)$$

$$\mathbb{C} = \{\mathbf{c}_w\}, 1 \leq w \leq W \quad (2.21c)$$

Coeficientes mel-cepstrais (MFCCs): palavra "Zero", 22.05 kHz, 8 bits

Magnitude $c(w,m)$

Coeficientes mel-cepstrais (MFCCs): palavra "Zero", 22.05 kHz, 8 bits

Magnitude $c(w,m)$

Figura 15: Coeficientes mel-cepstrais ($\mathbb{C}$) da palavra “zero”, $M = 16$ coeficientes

3 RECONHECIMENTO DE LOCUTOR POR *SELF-ORGANIZING MAPS*

3.1 Visão Geral

Após extraídos os vetores de atributos das locuções originais, estes são usados para treinar modelos conexionistas associados aos seus locutores. Cada locutor é representado por sua própria rede neural artificial de arquitetura *Self-Organizing Map* (SOM), treinada para executar a quantização vetorial dos vetores de MFCCs extraídos de suas próprias locuções. Quando apresentada uma nova locução (através de seus vetores de MFCCs), esta é testada em cada uma das SOMs, sendo declarada a vencedora aquela que apresenta a menor distorção total na quantização vetorial destes MFCCs. O locutor reconhecido é aquele associado à SOM vencedora.

Procedimentos semelhantes foram testados por Bennani e Gallinari (1990), Anderson e Patterson (1994) e Homayounpour e Chollet (1995), com bons resultados.

Neste capítulo são introduzidas as redes neurais artificiais e apresentadas as principais arquiteturas utilizadas para reconhecimento de locutor. A arquitetura *Self-Organizing Map* é então detalhada e por fim é explicado o processo de modelagem do locutor através destas redes.

3.2 Redes Neurais Artificiais

Segue a definição de Rede Neural Artificial (*Artificial Neural Network* – ANN), segundo Haykin (1999, pg. 2):

Uma rede neural é um processador distribuído altamente paralelo, composto de unidades de processamento simples, que são naturalmente propensas a armazenar conhecimento experimental e a torná-lo disponível para o uso. Ela se assemelha ao cérebro em dois aspectos:

1. O conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizado.
2. As forças das conexões entre neurônios, conhecidas como pesos sinápticos, são usadas para armazenar o conhecimento adquirido.

Redes neurais (BISHOP, 1995; ARBIB, 1995; HAYKIN, 1999) podem ser usadas como funções discriminantes não paramétricas em tarefas de classificação, como modelos de regressão múltipla ou como aproximadores de funções “universais”. Por estas propriedades, elas são aplicadas numa grande variedade de tarefas em reconhecimento de voz, seja como paradigma principal ou combinadas com outros modelos. O uso de redes neurais para a tarefa de reconhecimento de voz (KATAGIRI, 2000) é um campo de pesquisa com uma história de algumas décadas, envolvendo muitas disciplinas tais como a Fisiologia, a Física, a Estatística, a Psicologia, a Lingüística e a Engenharia.

3.2.1 Unidades de Processamento

A base do projeto de redes neurais artificiais é o modelo matemático do neurônio, que é chamado de Unidade de Processamento (*Processing Unit* – PU, *Processing Element* – PE). O modelo mais usado é o neurônio não linear, ilustrado na figura 16. Os três elementos básicos de uma unidade de processamento U são:

1. Um conjunto de *conexões* ou *sinapses*, cada uma das quais caracterizada por seu *peso*. A conexão $C_{k,i}$, que liga a unidade de processamento U_i à unidade U_k (nesta ordem) tem seu peso representado por $w_{k,i}$.
2. Um *somador* que faz a combinação linear dos sinais de entrada x_i , ponderados segundo os pesos das conexões $w_{k,i}$.
3. Uma *função de ativação* $\varphi(\cdot)$ para limitar a amplitude do sinal de saída da unidade de processamento y_k .

O modelo de neurônio da figura 16 também inclui um termo de *bias* b_k , aplicado externamente, que pode ser incorporado facilmente às entradas de U_k definindo-se uma entrada fictícia com as seguintes propriedades

$$x_0(t) = 1, \forall t \quad (3.1a)$$

$$w_{k,0}(t) = b_k(t) \quad (3.1b)$$

O conjunto de entradas da unidade U_k pode ser representado vetorialmente por

$$\mathbf{x}(t) = [x_0(t), x_1(t), \dots, x_i(t), \dots, x_I(t)]^T \quad (3.2)$$

e o conjunto de pesos das conexões com destino a U_k (conexões de entrada), no instante t em que é apresentada a entrada $\mathbf{x}(t)$, por

$$\mathbf{w}_k(t) = [w_{k,0}(t), w_{k,1}(t), \dots, w_{k,i}(t), \dots, w_{k,I}(t)]^T \quad (3.3)$$

onde I é o número de conexões com destino à unidade de processamento U_k .

Em termos matemáticos, a unidade de processamento U_k pode ser descrita pelo seguinte par de equações:

$$v_k(t) = \sum_{i=0}^I x_i(t) w_{k,i}(t) = \mathbf{x}^T(t) \mathbf{w}_k(t) \quad (3.4a)$$

$$y_k(t) = \varphi(v_k(t)) \quad (3.4b)$$

onde a variável $v_k(t)$ é chamada de *campo local induzido* pela apresentação da entrada $\mathbf{x}(t)$.

Figura 16: Modelo não linear de um neurônio

Funções de Ativação

As funções de ativação definem, segundo o modelo de neurônio linear exposto, o valor da saída em função do campo local induzido. As mais usadas são expostas a seguir, podendo ser encontradas com variações nos limites de definição.

1. Função Degrau

$$y_k = \varphi(v_k) = \begin{cases} 1 & , \text{ se } v_k \geq 0 \\ 0 & , \text{ se } v_k < 0 \end{cases} \quad (3.5)$$

2. Função Rampa

$$y_k = \varphi(v_k) = \begin{cases} 1 & , \text{ se } v_k \geq 1 \\ v_k & , \text{ se } 0 \leq v_k < 1 \\ 0 & , \text{ se } v_k < 0 \end{cases} \quad (3.6)$$

3. Função Sigmoidal

$$y_k = \varphi(v_k) = \frac{1}{1 + \exp(-av_k)} \quad (3.7)$$

onde a é o parâmetro de curvatura da função sigmoidal.

4. Função Tangente Hiperbólica

$$y_k = \varphi(v_k) = \tanh(v_k) \quad (3.8)$$

3.2.2 Arquiteturas de Redes

Por arquitetura de uma rede neural entende-se a forma como estão conectadas as suas unidades de processamento e como ocorre o fluxo do sinal dentro da rede. A arquitetura da rede neural está fortemente ligada ao algoritmo de aprendizado usado para treiná-la. As arquiteturas mais importantes são destacadas mais adiante neste capítulo.

Camadas

Usualmente, as unidades de processamento estão dispostas em *camadas (layers)*, sendo que unidades pertencentes a uma mesma camada apresentam propriedades semelhantes, tanto em termos de definição da unidade quanto de propagação do sinal.

A camada de índice ℓ será denotada por L_ℓ . Uma unidade de índice u pertencente a L_ℓ será denotada por U_u^ℓ . O número de unidades de L_ℓ será denotado por U_ℓ . A conexão da unidade $U_{u_1}^{\ell_1}$ para a unidade $U_{u_2}^{\ell_2}$ será denotada por $C_{u_2, u_1}^{\ell_2, \ell_1}$.

Quando todas as unidades de uma camada L_ℓ estão conectadas a todas as unidades da camada seguinte $L_{\ell+1}$ ($I_{\ell+1} = U_\ell, \forall U^{\ell+1}$), a rede é dita *totalmente conectada*; caso contrário a rede é *parcialmente conectada*.

Para um par de camadas L_1 e L_2 , pode-se definir a *matriz de conexão* $\mathbf{W}_{2,1}$ das

unidades de L_1 para as unidades de L_2 como

$$\mathbf{W}_{2,1} = [\mathbf{w}_1^{2,1}, \mathbf{w}_2^{2,1}, \dots, \mathbf{w}_{U_2}^{2,1}] = \quad (3.9a)$$

$$= \begin{bmatrix} w_{1,1}^{2,1} & w_{2,1}^{2,1} & \dots & w_{U_2,1}^{2,1} \\ w_{1,2}^{2,1} & w_{2,2}^{2,1} & \dots & w_{U_2,2}^{2,1} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1,U_1}^{2,1} & w_{2,U_1}^{2,1} & \dots & w_{U_2,U_1}^{2,1} \end{bmatrix} \quad (3.9b)$$

onde $\mathbf{w}_{u_2}^{2,1}$ é o vetor de pesos das conexões das unidades de L_1 com a unidade $U_{u_2}^2$ e $w_{u_2,u_1}^{2,1}$ é o peso da conexão $C_{u_2,u_1}^{2,1}$ da unidade $U_{u_1}^1$ para a unidade $U_{u_2}^2$.

Redes *Feedforward* de Uma Camada

A arquitetura mais simples de rede neural é composta por uma camada de entrada L_{in} , que se conecta unidirecionalmente para frente (*feedforward*) com uma camada de saída L_{out} (figura 17). Nesta arquitetura, a camada de entrada é composta por U_{in} unidades de processamento do tipo *pass-through*, que não executam qualquer processamento a não ser a distribuição do sinal de entrada para as U_{out} unidades da camada de saída.

Figura 17: Rede neural *feedforward* de uma camada

Redes *Feedforward* Multi-camada

Diferenciam-se das redes de uma camada pela presença de uma ou mais *camadas ocultas* L_{hid} , cujas unidades de processamento são chamadas de *unidades ocultas*

$U_{u_{hid}}^{hid}, 1 \leq u_{hid} \leq U_{hid}$ (figura 18). O acréscimo de uma ou mais camadas ocultas permite à rede extrair estatísticas de maior ordem, o que é particularmente útil quando a camada de entrada tem muitas unidades. A arquitetura da rede pode ser descrita sucintamente pela seqüência de números de unidades nas camadas, da entrada para a saída. A rede da figura 18, por exemplo, pode ser escrita como 10-4-2, significando $U_{in} = 10$ unidades na camada de entrada, $U_{hid} = 4$ na camada oculta e $U_{out} = 2$ na camada de saída.

Figura 18: Rede neural *feedforward* multi-camada, 10-4-2

Redes Recorrentes

Uma rede recorrente (*recurrent*) diferencia-se de uma rede *feedforward* pela presença de pelo menos um ciclo de realimentação (*feedback*). Uma conexão de realimentação conecta um neurônio de uma certa camada com um neurônio de uma camada anterior ou consigo mesmo. As redes recorrentes serão explicadas mais adiante na seção 3.3.4.

3.2.3 Processos de Aprendizado de Redes Neurais

Segue a definição de *aprendizado*, no contexto de redes neurais, segundo Haykin (1999, pg.50).

Aprendizado é um processo pelo qual os parâmetros livres de uma rede neural são adaptados, através de estímulos do ambiente onde está a rede neural. O tipo de aprendizado é determinado pela maneira pela qual as mudanças nos parâmetros acontecem.

Este processo é formalizado através de um *algoritmo de aprendizado*, que define como a rede é estimulada, como os parâmetros se adaptam e como a rede deve responder a novos estímulos.

Aprendizado Supervisionado

O conhecimento está disponível para a rede sob a forma de exemplos de pares $\mathbf{t}(t)$ do tipo vetor de entrada $\mathbf{x}(t)$ e seu respectivo vetor de saída desejada $\mathbf{d}(t)$, reunidos num conjunto de treinamento $\mathbb{T}$.

$$\mathbb{T} = \{\mathbf{t}(t)\} = \{(\mathbf{x}(t), \mathbf{d}(t))\}, 1 \leq t \leq T \quad (3.10)$$

onde T é o número de elementos do conjunto de treinamento. Costuma-se atribuir um significado temporal ao índice do elemento de treinamento, pois estes elementos são apresentados sequencialmente à rede neural para o seu treinamento. Neste contexto, uma apresentação do conjunto completo é chamada de *época* de treinamento, e T é a *duração da época de treinamento*.

Quando um exemplo é apresentado à rede neural, é calculado um sinal de erro $\mathbf{e}(t)$, no qual a rede se baseia para tentar apresentar uma resposta mais correta na próxima vez que o exemplo for apresentado.

$$\mathbf{e}(t) = \mathbf{d}(t) - \mathbf{y}(t) \quad (3.11)$$

Desta forma, pode-se entender que a rede aprende a imitar o seu conjunto de treinamento (ambiente).

Aprendizado Não Supervisionado

Neste caso, não há os vetores de saídas desejadas $\mathbf{d}(t)$ associados aos vetores de entradas $\mathbf{x}(t)$

$$\mathbb{T} = \{\mathbf{t}(t)\} = \{\mathbf{x}(t)\}, 1 \leq t \leq T \quad (3.12)$$

O sinal de erro gerado no aprendizado supervisionado é substituído por uma *me-*

dida independente da tarefa da qualidade da representação que a rede deve aprender, e os parâmetros livres são adaptados para minimizar este medidor. Para isto, pode ser usada uma regra de aprendizado competitivo.

Uma vez que a rede tenha sido sintonizada às regularidades estatísticas dos dados de entrada, ela desenvolve a habilidade de formar representações internas para a codificação dos atributos da entrada e criar novas classes automaticamente.

O aprendizado não supervisionado (auto-organizado) baseia-se em modificar repetidamente os pesos sinápticos de uma rede neural em resposta aos padrões de ativação, de acordo com regras pré-determinadas, até que uma determinada configuração final seja atingida.

Aprendizado Competitivo

No aprendizado competitivo (INTRATOR, 1995), as unidades de processamento de uma mesma camada empregam algum tipo de competição entre si através de conexões laterais. Esta idéia foi proposta originalmente por Rosenblatt (1958). As unidades individuais da rede aprendem a se especializar em conjuntos de padrões similares, tornando-se *detectores de atributos* para as diferentes classes de padrões de entrada.

Proposta inicialmente por Malsburg (1973), a competição *hard* define que apenas uma única unidade de saída pode estar ativa a cada instante. Segundo esta proposição, há três elementos básicos no aprendizado competitivo:

1. um conjunto de unidades de processamento idênticas, exceto por seus pesos sinápticos, respondendo de forma diferente a um conjunto de padrões de entrada;
2. um limite imposto à “força” de cada unidade;
3. um mecanismo que permite com que as unidades compitam pelo direito de responder a um determinado subconjunto de padrões de entrada, de forma que somente uma única unidade de saída, ou só uma unidade por grupo, esteja ativa num determinado instante.

Em modelos mais recentes, a competição afeta a atividade de unidades laterais, mas não necessariamente faz com que a ativação de todas as unidades, exceto uma, sejam levadas a zero (RUMELHART; ZIPSER, 1986). As arquiteturas que implementam este tipo de competição, chamado competição *soft*, podem possuir várias camadas. Suas unidades possuem conexões laterais inibitórias e conexões excitatórias com unidades de camadas adjacentes. Estes algoritmos de aprendizado competitivo podem ser

distinguidos por uma regra de aprendizado guiada por uma função objetiva desejada ou pela forma como a competição é promovida.

3.3 Redes Neurais para Reconhecimento de Locutor

A pesquisa recente demonstra que redes neurais constituem uma alternativa viável aos modelos estatísticos tradicionais em aplicações de reconhecimento automático de locutor (YUK, 1999). Elas se mostram particularmente adequadas para a exploração de características distintivas entre locutores.

Segue uma breve descrição das principais arquiteturas utilizadas em reconhecimento de locutor e suas aplicações.

3.3.1 *Multi-Layer Perceptron* (MLP)

As redes neurais de arquitetura *Multi-Layer Perceptron* (HAYKIN, 1999, cap. 4), treinadas pelo algoritmo *back-propagation* de aprendizado supervisionado (WERBOS, 1974; PARKER, 1985; LECUN, 1985; RUMELHART; HINTON; WILLIAMS, 1986a, 1986b) possuem uso generalizado. Derivadas dos *perceptrons* (MINSKY; PAPERT, 1990), possuem uma ou mais camadas intermediárias, compostas de unidades de processamento não lineares, que lhes conferem grande poder como classificadores de padrões, entre outras diversas aplicações.

Redes do tipo MLP, por não efetuarem processamento temporal, podem ser aplicadas em reconhecimento de locutor independente de texto. Neste caso, foram propostos modelos (OGLESBY; MASON, 1990) em que múltiplas MLPs se especializam em locutores específicos para a tarefa de identificação de locutor. Estudos sobre o uso de MLPs para a verificação de locutor podem ser encontrados em Tsoi (1994), Delacretaz e Hennebert (1998) e Sória (2001). Os pontos fracos das MLPs residem no tempo de treinamento muito grande e em dificuldades de convergência para grandes quantidades de dados.

3.3.2 *Self-Organizing Map* (SOM) e *Learning Vector Quantization* (LVQ)

As arquiteturas *Self-Organizing Map* e *Learning Vector Quantization*, são baseadas no processo de Quantização Vetorial (apresentado na seção 3.4) para implementar a detecção e o reconhecimento de padrões estatísticos. São redes neurais de uma única

camada, composta por unidades de processamento lineares, totalmente conectadas às unidades de entrada, onde são aplicados os vetores de entrada. Ambas são baseadas no processo de aprendizado competitivo (seção 3.2.3).

As SOM constituem a versão não supervisionada, e serão discutidas mais adiante na seção 3.5. Na arquitetura *Learning Vector Quantization* (LVQ) (KOHONEN, 1986), os pesos das unidades da camada de saída são associados aos *code vectors* representantes das classes. Quando é apresentado um padrão de uma entrada, uma unidade de saída é declarada vencedora, e seu vetor de pesos se aproxima ou distancia do padrão de entrada, de acordo com a classificação correta ou incorreta.

Aplicações de SOMs e LVQs para o reconhecimento de locutor em modo independente de texto podem ser encontradas em Bennani e Gallinari (1990), Anderson e Patterson (1994) e Homayounpour e Chollet (1995), com altos índices de reconhecimento. Mais recentemente, Lapidot, Guterman e Cohen (2002) usaram SOMs para descobrir automaticamente o número de locutores presentes num trecho longo de conversa, e então proceder a segmentação da conversa segundo os locutores descobertos.

3.3.3 Time-Delay Neural Network (TDNN)

A arquitetura *Time-Delay Neural Network*, descrita inicialmente por Lang e Hinton (1988) e Waibel (1989), se utiliza de *atrasos de tempo* para lidar com o processamento temporal. A TDNN é uma rede multi-camada, com propagação do sinal *feedforward*, cujas unidades ocultas e de saída são *replicadas no tempo*.

Figura 19: Rede neural TDNN para padrões de entrada unidimensionais

Redes TDNN foram utilizadas em tarefas de identificação de locutor com sucesso em Bennani e Gallinari (1991).

3.3.4 Recurrent Neural Network (RNN)

As redes neurais do tipo *recurrent* (DOYA, 1995) são aquelas que possuem conexões do tipo *feedback* e unidades que introduzem atrasos temporais no fluxo do sinal (figura 20). RNNs se comportam como sistemas dinâmicos: uma vez alimentadas com uma entrada, as conexões recorrentes são responsáveis pela evolução no tempo de seu estado interno. Desta forma, permitem que o estado de um sistema seja codificado nos padrões de atividade das unidades, possibilitando a programação de uma ampla variedade de comportamentos dinâmicos através dos pesos sinápticos.

Figura 20: Recurrent Neural Network (RNN)

Sua aplicação em um problema em modo dependente de texto, com bons resultados, pode ser encontrada em Tsoi (1994).

3.3.5 Neural Tree Network (NTN)

As *Neural Tree Networks* (SANKAR; MAMMONE, 1993) constituem um modelo híbrido de rede neural, que combinam árvores de decisão e redes neurais. As NTN são

classificadores hierárquicos que usam uma arquitetura em árvore para implementar uma estratégia de decisão sequencial linear. Em cada nó da rede, um *perceptron* é usado para dividir os dados para os nós seguintes.

NTNs foram aplicadas em experimentos de verificação de locutor independente de texto com bastante sucesso em Farrell, Mammone e Assaleh (1994).

3.4 Quantização Vetorial (*Vector Quantization* – VQ)

A *Quantização Vetorial* (*Vector Quantization* – VQ) é uma técnica que explora a estrutura subjacente de um conjunto de vetores de entrada com o objetivo de compressão dos dados (GERSHO; GRAY, 1992).

Sejam um vetor de entrada $\mathbf{x}(t) \in \mathbb{R}^n$ e um conjunto de unidades ou células, cada qual associada a um vetor de parâmetros (chamado *vetor de código*) $\mathbf{w}_u(t) \in \mathbb{R}^n$. A vencedora na categoria de problemas de VQ é definida como sendo a unidade cujo vetor de código possui a menor distância Euclidiana em relação a $\mathbf{x}(t)$:

$$u^*(t) = \arg \min_u \{\|\mathbf{x}(t) - \mathbf{w}_u(t)\|\} \quad (3.13)$$

Se $\mathbf{x}(t)$ é uma variável vetorial contínua, estocástica e natural não há a necessidade de se considerar múltiplos mínimos: a probabilidade de $\|\mathbf{x}(t) - \mathbf{w}_i(t)\| = \|\mathbf{x}(t) - \mathbf{w}_j(t)\|$ para $i \neq j$ é zero.

O objetivo dos métodos de VQ é o de gerar um conjunto de vetores de código que minimizem o erro de quantização:

$$E = \int \|\mathbf{x} - \mathbf{w}_{u^*}\|^2 p(\mathbf{x}) d\mathbf{x} = \min \quad (3.14)$$

onde $p(\mathbf{x})$ é a função de densidade de probabilidade de $\mathbf{x}$ e $d\mathbf{x}$ é um hipervolume diferencial no espaço de entradas. Observe-se que u^* , o índice da unidade vencedora, depende de $\mathbf{x}$ e de todos os vetores $\mathbf{w}_u$.

3.5 A Arquitetura *Self-Organizing Map* (SOM)

As redes do tipo *Self-Organizing Map* (KOHONEN, 2001) foram formalizadas por Kohonen (1990). Elas são baseadas no *aprendizado competitivo*, descrito na seção 3.2.3. Numa SOM, as unidades estão dispostas nos nós de uma retícula (*lattice*), normalmente uni- ou bidimensional (figura 21). As unidades são ativadas seletivamente

de acordo com os vários padrões de entrada (estímulos) ou com classes de padrões de entrada durante o processo de aprendizado competitivo. Esta ativação seletiva implementa a quantização vetorial dos vetores de entrada, descrito na seção 3.4: os vetores de pesos das unidades de saída constituem o conjunto de vetores de código.

A posição relativa das unidades da retícula, quando projetada no espaço de atributos, torna-se ordenada, de forma que um sistema de coordenadas significativo sobre os atributos de entrada é criado neste espaço. Desta forma, uma SOM se caracteriza pela formação de um *mapa topográfico* dos padrões de entrada, no qual as coordenadas das unidades são indicativos de atributos estatísticos intrínsecos contidos nestes padrões.

Figura 21: Arquitetura de uma *Self-Organizing Map*

Nesta arquitetura, as unidades são lineares e dispostas em uma única camada, totalmente conectadas aos nós de entradas com propagação *feedforward* dos sinais de entrada aplicados.

3.5.1 Algoritmo de Treinamento

A inicialização dos pesos sinápticos é feita através da atribuição de valores aleatórios pequenos, evitando a imposição de qualquer pré-ordenação. O algoritmo prossegue então em 3 etapas: competição, cooperação e adaptação, descritas a seguir.

Competição

Para cada padrão de entrada apresentado, cada unidade calcula seu valor de saída (valor da função discriminante). A distância Euclidiana entre o vetor de entrada e o

vetor de pesos da unidade é usada como função discriminante:

$$e_u(t) = \|\mathbf{x}(t) - \mathbf{w}_u(t)\| \quad (3.15)$$

Determina-se então a *unidade vencedora* U_{u^*} como sendo aquela que apresentou a menor distância do padrão de entrada:

$$u^*(t) = \arg \min_u e_u(t) \quad (3.16)$$

Alternativamente e de forma equivalente, o máximo produto interno entre o vetor de entrada e o vetor de pesos da unidade pode ser usado:

$$e_u(t) = \mathbf{w}_u(t) \cdot \mathbf{x}(t) \quad (3.17)$$

$$u^*(t) = \arg \max_u e_u(t) \quad (3.18)$$

Cooperação

A unidade vencedora determina em torno de si uma vizinhança topológica de unidades cooperativas. Para isso, escolhe-se inicialmente uma dimensão para a retícula (normalmente 1 ou 2), define-se o sistema de coordenadas nesta dimensão e uma métrica de distância entre unidades neste sistema (figura 22). Para os casos uni- e bidimensional, podem ser usadas as seguintes métricas:

$$d(U_{u1}, U_{u2}) = |u1 - u2|, \text{ para retícula unidimensional} \quad (3.19a)$$

$$d(U_{u1}, U_{u2}) = \sqrt{\|\mathbf{r}_{u1} - \mathbf{r}_{u2}\|}, \text{ para retícula bidimensional} \quad (3.19b)$$

Portanto, dada a unidade vencedora U_{u^*} , calcula-se inicialmente a distância entre ela e cada uma das demais $d(U_{u^*}, U_u), u \neq u^*$. A intensidade da interação entre a unidade vencedora e suas vizinhas é dada por uma *função de vizinhança* $h(d, t)$, com as seguintes propriedades:

- A função de vizinhança é simétrica em torno do ponto de máximo, definido em $u1 = u2$;
- A amplitude da função de vizinhança decresce monotonicamente com o aumento da distância lateral, decaindo para zero para $d \rightarrow \infty$.

Figura 22: Retícula bidimensional com sistema de coordenadas para as unidades U_{u1} e U_{u2}

A função *Gaussiana* é tipicamente usada, pois satisfaz ambas as condições e é *invariante à translação* (não depende da unidade central):

$$h(d, t) = \exp\left(-\frac{d^2}{2\rho^2(t)}\right) \quad (3.20)$$

onde $\rho(t)$ é a função *largura efetiva* da vizinhança topológica, que decai com o tempo, reduzindo a intensidade da interação cooperativa com as unidades vizinhas:

$$\rho(t) = \rho_0 \exp\left(-\frac{t}{\tau_1}\right) \quad (3.21)$$

com ρ_0 sendo a largura efetiva inicial (em $t = 0$) e τ_1 a constante de tempo de decaimento.

Assim, tendo calculado as distâncias entre a unidade vencedora e todas as demais unidades, calcula-se o valor da função de vizinhança para cada uma delas, determinando as intensidades de cooperação individuais.

Adaptação

Por fim, o processo de adaptação dos pesos sinápticos é executado, fazendo com o vetor de pesos da unidade vencedora seja movido em direção ao vetor de entradas atual.

As unidades vizinhas também têm seus pesos sinápticos adaptados, de acordo com o valor da função de vizinhança definido no processo cooperativo. Desta forma, a lei de atualização de pesos para a SOM é dada por:

$$\mathbf{w}_u(t+1) = \mathbf{w}_u(t) + \eta(t) h\left(d\left(\mathbf{U}_{u^*(t)}, \mathbf{U}_u\right), t\right) (\mathbf{x}(t) - \mathbf{w}_u(t)) \quad (3.22)$$

onde $\eta(t)$ é a *função taxa de aprendizado*, que começa com um valor inicial alto η_0 e decai exponencialmente com o tempo:

$$\eta(t) = \eta_0 \exp\left(-\frac{t}{\tau_2}\right) \quad (3.23)$$

O processo adaptativo da SOM pode ser decomposto em duas grandes etapas:

Auto-organização. Etapa inicial, em que a SOM parte de um estado de total desordem e chega ao estado de ordenação topológica. Nesta primeira etapa, que dura cerca de 1000 iterações, deve ser usada uma alta taxa de aprendizado e uma largura efetiva de vizinhança alta. Para a taxa de aprendizado, pode-se assumir uma taxa inicial $\eta_0 = 0.1$, decaindo para 0.01 após 1000 iterações ($\tau_2 = 1000$). Quanto à largura da vizinhança, deve incluir inicialmente quase todas as unidades da rede (ρ_0 é o “raio” da retícula), decaindo para somente algumas unidades vizinhas ou somente à unidade vencedora ao final. Para isto, pode-se usar a constante de tempo $\tau_1 = 1000/\log \rho_0$.

Convergência. Nesta segunda etapa, o processo adaptativo faz a sintonia fina do mapa de atributos, fazendo uma quantização estatística acurada do espaço de entradas. Como regra geral, a duração desta etapa é de aproximadamente 500 vezes o número de unidades que compõem o mapa. A taxa de aprendizado deve ser mantida fixa, com um valor baixo ($\eta = 0.01$). A vizinhança deve ser mantida constante, em aproximadamente uma ou nenhuma unidade vizinha.

3.5.2 Propriedades do Mapa de Atributos

Uma vez treinada, o mapa criado pela SOM evidencia características estatísticas importantes do espaço de entradas, que são descritas a seguir.

Aproximação do Espaço de Entradas. Os vetores de pesos das unidades da SOM formam um conjunto de protótipos que representam as características essenci-

ais do espaço de entradas, constituindo uma aproximação deste espaço. Esta característica é baseada na idéia de *quantização vetorial*.

Ordenação Topológica. O processo cooperativo do algoritmo de treinamento faz com que a posição relativa das unidades na retícula se torne uma evidência de semelhanças estatísticas entre as regiões do espaço de entradas codificadas por estas unidades.

Mapeamento de Densidades de Probabilidades. Mais unidades têm seus vetores de pesos associados a regiões do espaço de entradas com maior probabilidade de ocorrência. Isto significa que o conjunto de vetores de pesos têm maior resolução em regiões do espaço de entradas mais ricas em informações estatísticas.

Seleção de Atributos. Dados padrões de entrada com uma distribuição não linear, a SOM é capaz de selecionar o melhor conjunto de atributos para aproximar esta distribuição. Mais precisamente, os mapas fornecem uma aproximação discreta das *curvas principais* ou *superfícies principais*, e podem ser tomados como uma generalização não linear da análise de componentes principais.

3.6 *Self-Organizing Maps* para Reconhecimento Automático de Locutor

3.6.1 Princípio de Funcionamento

Para executar a tarefa de reconhecimento, a voz de cada locutor é modelada por uma rede neural de arquitetura SOM. As SOMs de todos os locutores são idênticas entre si quanto ao número total de unidades, tamanho e disposição das retículas e parâmetros de treinamento.

Cada SOM é treinada somente com vetores de MFCCs extraídos de locuções de seu próprio locutor, tornando-se especialista na quantização destes vetores. Assim, se estes vetores contêm características essenciais suficientes do locutor, uma SOM apresentará um erro total de quantização menor quando apresentada a uma locução de seu locutor do que quando apresentada a uma locução de outro locutor.

Quando uma nova locução é apresentada, as SOMs competem entre si, sendo declarada vencedora aquela que fornecer o menor erro de quantização total para esta locução. O locutor associado à rede vencedora é considerado o locutor reconhecido.

3.6.2 Normalização

Antes de serem submetidos às SOMs, os conjuntos de vetores de atributos das locuções devem ser *normalizados*. Esta operação se justifica pelas diferenças entre as médias e variâncias entre as distribuições dos $2M$ MFCCs e delta-MFCCs: pequenas variações ocorridas em coeficientes que variam pouco podem ser mais significativas do que grandes variações em coeficientes que variam muito, levando a métricas erradas no cálculo da distância entre vetores de atributos (equação 3.15).

Seja o conjunto de treinamento $\mathbb{C}_l = \{\mathbf{c}_{w,l,p}\}$, o conjunto de vetores de MFCCs, extraídos dos segmentos das locuções $\mathbf{X}_{l,p}$ do locutor l , onde p é a frase ou palavra emitida na locução ($1 \leq p \leq P$) e w é o índice do segmento ($1 \leq w \leq W_{l,p}$). Sem perda de generalidade, assumamos que apenas uma repetição de cada frase p existe no conjunto de treinamento ($I = 1$).

O processo de normalização consiste em subtrair a média ($\mu_{c_{m,l,p}}$) e dividir pelo desvio padrão ($\sigma_{c_{m,l,p}}$) cada um dos componentes dos vetores de atributos, calculados sobre cada locução p do locutor l .

$$\mu_{c_{m,l,p}} = \frac{1}{W_{l,p}} \sum_{w=1}^{W_{l,p}} c_{m,w,l,p} \quad (3.24a)$$

$$\sigma_{c_{m,l,p}} = \sqrt{\frac{1}{W_{l,p}} \sum_{w=1}^{W_{l,p}} [c_{m,w,l,p} - \mu_{c_{m,l,p}}]^2} \quad (3.24b)$$

$$\hat{c}_{m,w,l,p} = \frac{c_{m,w,l,p} - \mu_{c_{m,l,p}}}{\sigma_{c_{m,l,p}}} \quad (3.24c)$$

$$\hat{\mathbf{c}}_{w,l,p} = \left[\hat{c}_{m,w,l,p} \right]_{m=1}^{2M} \quad (3.24d)$$

Procedimento idêntico se aplica às locuções individuais do conjunto de testes.

3.6.3 Treinamento

Seja $\mathbb{L}$ um conjunto de L locutores, considerados alvos do reconhecimento. Tome-se l , um locutor qualquer pertencente a $\mathbb{L}$. Seja $\mathbb{C}_l$ o conjunto de treinamento do locutor l , definido na seção anterior.

Para cada locutor, define-se uma rede neural $\mathbb{N}_l$, com arquitetura SOM, que é treinada com $\mathbb{C}_l$ de acordo com o algoritmo de treinamento definido na seção 3.5.1. Ao final do processo de treinamento, os vetores de pesos $\mathbf{w}_{u,l}$ das unidades de saída de $\mathbb{N}_l$ representam os *code vectors* do locutor l , que minimizam o erro total de quantização

dos segmentos de locução que compõem $\mathbb{C}_l$:

$$E_l = E(\mathbb{N}_l) = \sum_{p=1}^P \sum_{w=1}^{W_{l,p}} \|\hat{\mathbf{c}}_{w,l,p} - \mathbf{w}_{u^*,l}\| \quad (3.25)$$

onde u^* é o índice da unidade vencedora para o segmento de locução.

3.6.4 Teste

Seja uma nova locução $\mathbf{X}_{\bar{l},\bar{p}}$, de uma frase $\bar{p}$ desconhecida, emitida por um locutor $\bar{l}$ desconhecido, representada pelos seus vetores de MFCCs e delta-MFCCs normalizados $\{\hat{\mathbf{c}}_{w,\bar{l},\bar{p}}\}$. O número de segmentos $W_{\bar{l},\bar{p}}$ é conhecido, pois depende apenas da duração da locução e dos parâmetros do extrator de atributos.

Esta locução é apresentada a cada rede $\mathbb{N}_l$, sendo calculado o erro total de quantização desta locução em cada rede:

$$E(\mathbb{N}_l) = \sum_{w=1}^{W_{\bar{l},\bar{p}}} \|\hat{\mathbf{c}}_{w,\bar{l},\bar{p}} - \mathbf{w}_{u^*,l}\| \quad (3.26)$$

Através de um processo competitivo semelhante ao das unidades dentro de cada SOM, é declarada vencedora a rede $\mathbb{N}_{l^*}$ que apresenta o mínimo erro total de quantização

$$E(\mathbb{N}_{l^*}) = \min_l E(\mathbb{N}_l) \quad (3.27)$$

e o locutor reconhecido será aquele associado a $\mathbb{N}_{l^*}$

$$l^* = \arg \min_l E(\mathbb{N}_l) \quad (3.28)$$

Desta forma, se $l^* = \bar{l}$, o locutor foi corretamente identificado.

4 PROCEDIMENTO EXPERIMENTAL

A tarefa escolhida como aplicação experimental foi a de identificação de locutor, em um conjunto fechado de locutores, em modo independente de texto. Seguem os detalhes sobre o *corpus* de locuções utilizados, as arquiteturas experimentadas e os resultados obtidos.

4.1 *Corpus* de Amostras de Voz

Um novo *corpus* de amostras de vozes foi coletado para a tarefa. Em resumo, o conjunto consiste de frases em Português, faladas por 14 locutores brasileiros (6 homens e 8 mulheres), com 32 frases por locutor, sendo 26 frases comuns para todos os locutores e 6 frases variáveis por locutor. Todas as amostras foram coletadas no mesmo local, na mesma data, em uma única sessão. Não foi executado qualquer processamento para a eliminação de ruídos ou silêncios sobre as amostras coletadas.

4.1.1 Condições do Ambiente

A coleta foi executada num ambiente silencioso e de pouca reverberação. Foi utilizado um microfone de baixo custo *Plantronics Audio 90*, de uso comum em computadores pessoais e *videogames* que usam reconhecimento de voz. Este microfone, montado num conjunto do tipo *headset*, permite que seja mantida uma distância fixa entre o rosto do locutor e o microfone. O microfone foi posicionado a uma distância de aproximadamente 4cm do rosto, numa posição intermediária entre o nariz e a boca.

4.1.2 Composição do Conjunto de Locutores

O conjunto de locutores (tabela 1) foi composto por 14 locutores brasileiros, nascidos no estado de São Paulo, sendo 6 homens e 8 mulheres, na faixa de idade entre 28 e 42 anos (média aproximada de 32 anos).

ID (I)	Nome	Sexo	Idade	Cidade Natal
01	Alexandre T. Mafra	M	31	São Paulo
02	Lúcia T. G. de Castro	F	42	Santos
03	Renata R. Rodrigues	F	32	Campinas
04	Mauro A. DiGiorgi	M	30	São Paulo
05	Eduardo A. C. Fullen	M	30	São Caetano do Sul
06	Alessandra M. Velho	M	28	Campinas
07	Andréia A. C. Martins	F	31	São Paulo
08	Claudia M. S. Bernasconi	F	33	São Paulo
09	Maurício B. Falleiros	M	32	São Paulo
10	Ana Paula Henriques	F	30	São Paulo
11	Roberto C. Godoy	M	32	São Paulo
12	Paulo Sergio S. Pfaff	M	38	São Paulo
13	Gabriela S. Ferreira	F	30	Ribeirão Preto
14	Fernanda Henriques	M	28	São Paulo

Tabela 1: Composição do conjunto de locutores do *corpus* de amostras de voz

4.1.3 Frases Usadas

Foram coletadas, para cada locutor, 32 frases em Português. Destas, 6 foram frases variáveis por locutor (conjunto VAR1, tabela 2) e as outras 26 foram frases idênticas para todos os locutores. Estas 26 frases se agrupam em 3 conjuntos foneticamente balanceados, extraídos do trabalho de Alcaim, Solewicz e Moraes (1992): um primeiro menor, com 6 frases (conjunto BAL1, tabela 3), e dois outros maiores, cada um com 10 frases (conjuntos BAL2 e BAL3, tabelas 4 e 5, respectivamente).

O objetivo do uso conjuntos de frases foneticamente balanceadas é o de se obter, para cada locutor, amostras dos fones que compõem a língua falada, na proporção em que ocorrem nesta língua. Com isto, aumenta-se a representatividade estatística das amostras coletadas, quando as frases são usadas em conjunto para o treinamento do reconhecedor.

O uso de frases variáveis por locutor tem o objetivo de testar o desempenho do reconhecimento com frases que poderiam ser usadas como teste em aplicações comerciais. Usadas como conjunto de treinamento, podem servir como comparativo entre o uso ou não de frases foneticamente balanceadas.

VAR1	
ID (p)	Frase
001	Diga seu nome completo
002	Diga sua cidade natal
003	Diga a data o seu nascimento ¹
004	Diga o CEP de onde mora ²
005	Diga que horas são ³
006	Diga sua comida favorita

¹ Ex.: dez de abril de mil novecentos e setenta e um

² Ex.: zero quatro zero sete sete zero zero zero

³ Ex.: são dezenove horas e catorze minutos

Tabela 2: Conjunto VAR1 de frases variáveis por locutor

BAL1	
ID (p)	Frase
101	Eu vi logo a Ioiô e o Léo.
102	Um homem não caminha sem um fim.
103	Vi Zé fazer essas viagens seis vezes.
104	O atabaque do Tito é coberto com pele de gato.
105	Ele lê no leito de palha.
106	Paira um ar de arara rara no Rio Real.

Tabela 3: Conjunto BAL1, foneticamente balanceado, de frases comuns a todos os locutores

BAL2	
ID (p)	Frase
201	Nosso telefone quebrou.
202	Desculpe se magoei o velho.
203	Queremos discutir o orçamento.
204	Ela tem muita fome.
205	Uma índia andava na mata.
206	Zé, vá mais rápido!
207	Hoje dormirei bem.
208	João deu pouco dinheiro.
209	Ainda são seis horas.
210	Ela saía discretamente.

Tabela 4: Conjunto BAL2, foneticamente balanceado, de frases comuns a todos os locutores

BAL3	
ID (p)	Frase
301	Um casal de gatos come no telhado.
302	A cantora foi apresentar seu grande sucesso.
303	Lá é um lugar ótimo para tomar uns chopinhos.
304	O musical consumiu sete meses de ensaio.
305	Nosso baile inicia após as nove.
306	Apesar desses resultados, tomarei uma decisão.
307	A verdade não poupa nem as celebridades.
308	As queimadas devem diminuir este ano.
309	O vão entre o trem e a plataforma é muito grande.
310	Infelizmente não compareci ao encontro.

Tabela 5: Conjunto BAL3, foneticamente balanceado, de frases comuns a todos os locutores

4.2 *Software e Hardware Usados*

Todo o *software* utilizado, tanto para o Extrator de Atributos quanto para as SOMs, foi desenvolvido em ANSI-C, rodando sobre o sistema operacional *RedHat Linux 7.1*. Quanto ao *hardware*, foi usado um PC com processador *Pentium III*, com frequência de *clock* de 500 MHz, equipado com 256 Mb de memória RAM e placa de som comum para PC modelo *Creative Soundblaster AWE 64*, usada para a coleta e digitalização das amostras de voz.

4.3 Digitalização e Extração dos MFCCs

As vozes foram coletadas a uma taxa de amostragem $f_c = 22.05kHz$, com resolução de 8 bits e armazenadas em arquivos de formato *.wav*. Para a pré-ênfase, foi usado $a = 0.95$.

As amostras foram então particionadas em segmentos de $23.22ms$ ($N_w = 512$ amostras), sobrepostas em $11.61ms$ ($s = 0.5$, 256 amostras), sobre os quais foi aplicada a janela de Hamming. Foi usado um banco Mel-Frequency de 32 filtros triangulares, com frequências centrais igualmente espaçadas por $120mel$, sendo a primeira delas igual a $480mel$. Foram usados os 14 primeiros MFCCs, cobrindo aproximadamente o espectro entre $263Hz$ e $4592Hz$. A estes 14 MFCCs, foram acrescentados seus 14 delta-MFCCs, compondo vetores de dimensão $2M = 28$.

A tabela 6 apresenta as características dos conjuntos que compõem o *corpus*, após

Característica	VAR1	BAL1	BAL2	BAL3
Total de frases	84	84	140	140
Frases por locutor	6	6	10	10
Duração total	223.26 s	244.04 s	281.89 s	419.58 s
Duração mínima de um locutor	13.12 s	14.11 s	14.51 s	24.85 s
Duração média por locutor	15.95 s	17.43 s	20.14 s	29.97 s
Duração máxima de um locutor	20.43 s	21.19 s	26.01 s	36.34 s
Duração mínima de uma locução	0.87 s	1.57 s	1.03 s	2.03 s
Duração média de uma locução	2.66 s	2.91 s	2.01 s	3.00 s
Duração máxima de uma locução	6.04 s	4.46 s	2.96	4.35 s
Total de segmentos	19230	21020	24280	36140
Mínimo de segmentos por locutor	1130	1215	1250	2140
Média de segmentos por locutor	1374	1501	1734	2581
Máximo de segmentos por locutor	1760	1825	2240	3130
Mínimo de segmentos por locução	75	135	95	175
Média de segmentos por locução	229	250	173	258
Máximo de segmentos por locução	520	384	255	375

Tabela 6: Características dos conjuntos VAR1, BAL1, BAL2 E BAL3, após a digitalização e extração de atributos

a digitalização e extração dos MFCCs.

4.4 Arquiteturas de SOMs Usadas

Foram testadas 4 arquiteturas de SOMs, que se distinguem primariamente pelo número de unidades de saída: 16 (SOM 16), 25 (SOM 25), 36 (SOM 36) e 64 (SOM 64). Em todas elas, foram usadas retículas bidimensionais quadradas. Os vetores de pesos foram inicializados, em todos os casos, com uma variável aleatória de distribuição Gaussiana, com média $\mu = 0$ e desvio padrão $\sigma = 0.005$.

4.5 Treinamento

Em cada rodada, foram treinadas 14 SOMs (uma para cada locutor), com uma das arquiteturas anteriores (SOM 16, SOM 25, SOM 36 ou SOM 64), sobre um dos conjunto de palavras (VAR1, BAL1 e BAL2). O conjunto BAL3 não participou de nenhuma rodada de treinamento, sendo usado apenas como referência para comparação dos resultados dos testes. Os vetores de MFCCs foram apresentados em ordem aleatória para cada rede. O treinamento foi interrompido quando um número máximo de épocas

pré-definido foi atingido. Os parâmetros de treinamento foram fixados por arquitetura, e estão relacionados na tabela 7.

Arquitetura	Unidades	Retícula	η_0	τ_2	ρ_0	τ_1	Épocas
SOM 16	16	4 x 4	0.1	400	4	40	150
SOM 25	25	5 x 5	0.1	1000	5	20	150
SOM 36	36	6 x 6	0.1	1000	6	20	100
SOM 64	64	8 x 8	0.1	1000	8	30	100

Tabela 7: Arquiteturas de SOM utilizadas e seus parâmetros de treinamento

Figura 23: Evolução do erro de quantização por segmento para cada arquitetura (SOM 16, SOM 25, SOM 36 e SOM 64) durante o treinamento do locutor $l = 1$ com o conjunto VAR1, segundo os parâmetros de treinamento definidos na tabela 7

Os valores utilizados para a constante τ_1 , que define a taxa de diminuição temporal da largura da vizinhança durante o processo colaborativo do treinamento, foram bastante inferiores aos recomendadas na seção 3.5.1. Isto foi possível porque observou-se em experimentos anteriores com arquiteturas fixas que, para os conjuntos de treinamento utilizados, valores altos para esta constante não melhoravam o valor final do erro de quantização, apenas tornando mais lento o processo de convergência.

Desta forma, a etapa de auto-organização das SOMs pode ser acelerada, passando rapidamente para a etapa de convergência. Isto permitiu a redução do número de épocas

de treinamento para 100 ou 150, também bastante inferior ao recomendado na seção 3.5.1.

As tabelas 8, 9, 10 e 11 mostram os resultados do treinamento para cada combinação de arquitetura, conjunto de treinamento e locutor. Estão tabelados os números de segmentos que compõem cada particular conjunto de treinamento (#segm), o erro médio por segmento ao final do treinamento $E(l)$ e a duração total do treinamento em segundos $t(s)$ obtida com o *software* e *hardware* descritos na seção 4.2. Ao final de cada tabela, são exibidos a média (μ) e o desvio padrão (ρ) entre os locutores, calculados para cada coluna, sem ponderação.

4.5.1 Análise dos Resultados do Treinamento

Conforme se observa na figura 23, arquiteturas com maior número de unidades convergem para um erro final de quantização menor. Este resultado corresponde às expectativas sobre a distorção total de um processo de quantização vetorial com o aumento do número de *code vectors*, detalhado em Deller Jr., Hansen e Proakis (1993, cap.7).

Os valores médios do erro de quantização entre os locutores $\mu_{E(l)}$ das tabelas 8, 9, 10 e 11 foram reunidos e plotados no gráfico da figura 24. Ainda na mesma figura, observar que o conjunto BAL 1 sempre atinge erros de quantização maiores ao final do processo de treinamento, seguido pelos conjuntos BAL2 e VAR1.

4.6 Resultados dos Testes

Para cada par composto por uma arquitetura e um conjunto de treinamento, foram testados todos os conjuntos, incluindo o teste sobre o próprio conjunto de treinamento. Os resultados destes testes podem ser verificados nas tabelas 12, 13, 14 e 15.

4.6.1 Análise dos Resultados dos Testes

O primeiro aspecto a ser analisado é a influência do conjunto de treinamento sobre os resultados dos testes. Tomando-se os resultados dos testes sobre o conjunto BAL3 como referência, a taxa de acerto para a identificação do locutor é superior quando as redes são treinadas com conjuntos maiores. Para demonstrar isto, o gráfico da figura 25 apresenta as taxas de erro de identificação sobre o conjunto BAL3 em função da duração total do conjunto de treinamento: a linha média indica a influência positiva da

SOM 16									
l	VAR1			BAL1			BAL2		
	#segm.	$E(l)$	$t(s)$	#segm.	$E(l)$	$t(s)$	#segm.	$E(l)$	$t(s)$
01	1130	3.850350	145	1275	3.930231	160	1425	3.907387	189
02	1510	3.756444	192	1610	3.888117	204	2065	3.854281	308
03	1205	3.737171	158	1480	3.892291	191	1660	3.862804	254
04	1440	3.704218	188	1545	3.776099	200	1755	3.740991	229
05	1340	3.711907	168	1465	3.828838	187	1615	3.761101	252
06	1705	3.724510	215	1825	3.829321	234	2240	3.834785	436
07	1760	3.705028	216	1680	3.851719	211	2055	3.783249	289
08	1270	3.710843	164	1270	3.842659	162	1490	3.799229	223
09	1140	3.821485	156	1215	3.926001	160	1250	3.922188	185
10	1155	3.721538	147	1300	3.877294	174	1560	3.781904	205
11	1360	3.809140	171	1600	3.896933	213	1710	3.905970	239
12	1465	3.780217	196	1705	3.880631	219	2000	3.803833	276
13	1405	3.742413	183	1570	3.853733	201	1955	3.773800	260
14	1345	3.689333	171	1480	3.852578	193	1500	3.842279	228
μ	1374	3.747471	176	1501	3.866175	194	1734	3.826700	255
ρ	168	0.042635	20	160	0.035267	20	249	0.050932	54

Tabela 8: Resultados do treinamento da arquitetura SOM 16 sobre os conjuntos VAR1, BAL1 e BAL2, para cada um dos locutores

SOM 25									
l	VAR1			BAL1			BAL2		
	#segm.	$E(l)$	$t(s)$	#segm.	$E(l)$	$t(s)$	#segm.	$E(l)$	$t(s)$
01	1130	3.638926	229	1275	3.733778	255	1425	3.708895	274
02	1510	3.555540	316	1610	3.692731	343	2065	3.651145	383
03	1205	3.517757	256	1480	3.694164	312	1660	3.650260	317
04	1440	3.496305	271	1545	3.577963	292	1755	3.542798	334
05	1340	3.483513	260	1465	3.621988	287	1615	3.574476	307
06	1705	3.510487	335	1825	3.620634	341	2240	3.632639	423
07	1760	3.507588	367	1680	3.643409	329	2055	3.586185	388
08	1270	3.492126	258	1270	3.625308	243	1490	3.578020	278
09	1140	3.606436	239	1215	3.711298	242	1250	3.709633	226
10	1155	3.496430	253	1300	3.652692	249	1560	3.571513	280
11	1360	3.603729	282	1600	3.698144	312	1710	3.708936	312
12	1465	3.553793	308	1705	3.676520	328	2000	3.608054	366
13	1405	3.530000	279	1570	3.657584	309	1955	3.588535	357
14	1345	3.482920	292	1480	3.638358	290	1500	3.634082	274
μ	1374	3.533968	282	1501	3.660326	295	1734	3.624655	323
ρ	168	0.043520	33	160	0.036620	31	249	0.048659	48

Tabela 9: Resultados do treinamento da arquitetura SOM 25 sobre os conjuntos VAR1, BAL1 e BAL2, para cada um dos locutores

SOM 36									
l	VAR1			BAL1			BAL2		
	#segm.	$E(l)$	$t(s)$	#segm.	$E(l)$	$t(s)$	#segm.	$E(l)$	$t(s)$
01	1130	3.492952	203	1275	3.589477	243	1425	3.565388	258
02	1510	3.404846	271	1610	3.560843	315	2065	3.524513	376
03	1205	3.367438	218	1480	3.560192	297	1660	3.513186	302
04	1440	3.369601	257	1545	3.427927	315	1755	3.393654	320
05	1340	3.342476	239	1465	3.489934	270	1615	3.435144	295
06	1705	3.353831	307	1825	3.487980	373	2240	3.490784	408
07	1760	3.355704	317	1680	3.499603	322	2055	3.460366	374
08	1270	3.344298	227	1270	3.474160	273	1490	3.433771	271
09	1140	3.458565	205	1215	3.550769	234	1250	3.577610	228
10	1155	3.365070	208	1300	3.511152	226	1560	3.432277	285
11	1360	3.460631	246	1600	3.564048	281	1710	3.571059	313
12	1465	3.413179	265	1705	3.538540	301	2000	3.466402	368
13	1405	3.379473	255	1570	3.518382	278	1955	3.474470	358
14	1345	3.350826	243	1480	3.503370	267	1500	3.508964	275
μ	1374	3.389921	247	1501	3.519741	285	1734	3.489113	317
ρ	168	0.041913	30	160	0.037235	34	249	0.050061	46

Tabela 10: Resultados do treinamento da arquitetura SOM 36 sobre os conjuntos VAR1, BAL1 e BAL2, para cada um dos locutores

SOM 64									
l	VAR1			BAL1			BAL2		
	#segm.	$E(l)$	$t(s)$	#segm.	$E(l)$	$t(s)$	#segm.	$E(l)$	$t(s)$
01	1130	3.257061	386	1275	3.369882	453	1425	3.354250	495
02	1510	3.196165	513	1610	3.338552	553	2065	3.325137	703
03	1205	3.173336	412	1480	3.324057	485	1660	3.321978	557
04	1440	3.163855	493	1545	3.206206	505	1755	3.188100	602
05	1340	3.132561	461	1465	3.252636	483	1615	3.231621	578
06	1705	3.163045	583	1825	3.274538	614	2240	3.292596	817
07	1760	3.154074	641	1680	3.288615	567	2055	3.259534	730
08	1270	3.110839	422	1270	3.231460	438	1490	3.209353	525
09	1140	3.228385	385	1215	3.318852	449	1250	3.349652	445
10	1155	3.128878	540	1300	3.290218	460	1560	3.233025	547
11	1360	3.235732	688	1600	3.346527	583	1710	3.350150	578
12	1465	3.198592	738	1705	3.328344	638	2000	3.282925	676
13	1405	3.183600	647	1570	3.323647	570	1955	3.258399	660
14	1345	3.126672	500	1480	3.303251	502	1500	3.305747	503
μ	1374	3.175200	529	1501	3.299770	521	1734	3.283033	601
ρ	168	0.038801	104	160	0.039551	58	249	0.047592	90

Tabela 11: Resultados do treinamento da arquitetura SOM 64 sobre os conjuntos VAR1, BAL1 e BAL2, para cada um dos locutores

SOM 16					
Conjunto de Treinamento	VAR1 P = 6	BAL1 P = 6	BAL2 P = 10	BAL3 P = 10	μ
VAR1	96.43%	85.71%	83.57%	93.57%	89.51%
BAL1	78.57%	100.00%	87.86%	93.57%	90.18%
BAL2	89.29%	90.48%	100.00%	97.14%	95.31%
μ	88.10%	92.06%	90.48%	94.76%	91.67%

Tabela 12: Porcentagens de identificação correta do locutor para a SOM 16

SOM 25					
Conjunto de Treinamento	VAR1 P = 6	BAL1 P = 6	BAL2 P = 10	BAL3 P = 10	μ
VAR1	100.00%	84.52%	82.14%	93.57%	89.50%
BAL1	85.71%	100.00%	93.57%	97.14%	94.42%
BAL2	88.10%	89.29%	100.00%	97.86%	95.10%
μ	91.27%	91.27%	91.90%	96.19%	93.01%

Tabela 13: Porcentagens de identificação correta do locutor para a SOM 25

SOM 36					
Conjunto de Treinamento	VAR1 P = 6	BAL1 P = 6	BAL2 P = 10	BAL3 P = 10	μ
VAR1	100.00%	86.90%	87.86%	95.71%	92.41%
BAL1	84.52%	100.00%	92.86%	99.29%	94.64%
BAL2	89.29%	90.48%	100.00%	97.86%	95.54%
μ	91.27%	92.46%	93.57%	97.62%	94.20%

Tabela 14: Porcentagens de identificação correta do locutor para a SOM 36

SOM 64					
Conjunto de Treinamento	VAR1 P = 6	BAL1 P = 6	BAL2 P = 10	BAL3 P = 10	μ
VAR1	100.00%	84.52%	87.86%	97.14%	92.41%
BAL1	86.90%	100.00%	93.57%	99.29%	95.31%
BAL2	88.10%	90.48%	100.00%	99.29%	95.76%
μ	91.67%	91.67%	93.81%	98.57%	94.49%

Tabela 15: Porcentagens de identificação correta do locutor para a SOM 64

Figura 24: Média do erro de quantização por segmento ao final do treinamento, calculada entre os locutores, em função do número de unidades da arquitetura, para cada um dos conjuntos de treinamento

duração sobre a queda da taxa de erros de identificação.

A segunda questão se refere ao poder discriminante de cada arquitetura. Usando ainda os resultados sobre o conjunto BAL3 como referência, foi levantado o gráfico da figura 26, que exibe a taxa de erro de identificação em função do número de unidades da arquitetura, para cada conjunto de treinamento. Observa-se, pela curva da média ponderada segundo o número de frases do conjunto de treinamento, a influência positiva do aumento do número de unidades da arquitetura, baixando as taxas de erro de identificação.

Por fim, são analisados os resultados das taxas de identificação em função do conjunto testado. Para os casos em que o conjunto de teste é o mesmo que o de treinamento, uma taxa de acerto esperada de 100% foi obtida em todos os casos, exceto para o par SOM 16 conjunto de treinamento VAR1, que foi de 96.43% (tabela 12), que pode ser considerada incapaz, com os parâmetros de treinamento usados, de quantizar adequadamente o conjunto. Para efeito de análise, os testes efetuados sobre os próprios conjuntos de treinamento serão descartados.

Observando as tabelas de resultados 12, 13, 14 e 15, verifica-se que as piores taxas de acerto ocorrem quando o conjunto VAR1 é testado, independentemente da arquitetura. Para explicar os motivos, serão detalhados os erros por frase ocorridos no teste do

Figura 25: Taxa de erro de identificação em função da duração do conjunto de treinamento

Figura 26: Taxa de erro de identificação em função do número de unidades da arquitetura usada, para cada conjunto de treinamento

conjunto VAR1 sobre a arquitetura SOM 64 treinada com o conjunto BAL1. A escolha se justifica pela aparente incompatibilidade entre os bons resultados da combinação no teste do conjunto BAL3 (99.29% de acerto) contra os resultados ruins no teste do conjunto VAR1 (86.90% de acerto). O alto índice de erros de classificação tende a evidenciar melhor os seus motivos.

Note-se inicialmente que o conjunto VAR1, tomado como conjunto de testes, não apresenta nenhuma propriedade relevante que demonstre que as frases possam ser analisadas em conjunto. A principal diferença entre as respostas dos locutores no conjunto VAR1 está na duração das respostas. A tabela 16 e o respectivo gráfico (figura 27) ilustram a variação da taxa de erros de identificação em função da duração média das frases, calculadas entre os locutores.

O gráfico demonstra de forma bastante clara que existe um limiar entre 2.45s e 2.79s para a duração média das locuções, acima do qual a taxa de acerto é de 100%. Mais do que isto, demonstra que, abaixo deste limiar, a redução da duração média das locuções faz com que as taxas de erro de identificação do locutor aumentem muito rapidamente. Verifica-se, portanto, que a duração das locuções de teste é um fator de alta relevância para este sistema de reconhecimento automático de locutor.

Frase (p)	Duração Média	Desv. Padr. Duração	Erros	Taxa de Acertos
001	2.45 s	0.47 s	1	92.86%
002	1.44 s	0.43 s	4	71.43%
003	3.43 s	0.51 s	0	100.00%
004	4.44 s	0.77 s	0	100.00%
005	2.79 s	0.36 s	0	100.00%
006	1.40 s	0.24 s	6	64.29%

Tabela 16: Taxa de erro de identificação em função da duração média da locução para a arquitetura SOM 64, treinada com o conjunto BAL1, testada com o conjunto VAR1

Figura 27: Duração média das locuções e taxas de erro de identificação para a arquitetura SOM 64, treinada com o conjunto BAL1, testada com o conjunto VAR1, segundo os dados da tabela 16

5 CONCLUSÕES

5.1 Conclusões Gerais

Neste trabalho foi apresentado inicialmente um conjunto de técnicas envolvendo a extração de atributos representativos de um locutor a partir de segmentos de amostras de sua voz. Foi descrita a modelagem dos locutores através de redes neurais de arquitetura *Self-Organizing Map* (SOM), treinadas exclusivamente com os atributos extraídos de seus locutores associados. A técnica de identificação apresentada consiste na comparação entre os erros de quantização vetorial das locuções de teste sobre as diversas SOMs, sendo o locutor identificado aquele cuja rede melhor quantiza a locução de teste.

Para testar a solução, foi proposto um problema de identificação de locutor em um conjunto fechado de locutores, em modo independente de texto. Para isto, foi coletado um *corpus* de amostras de voz de 14 locutores (6 homens e 8 mulheres), envolvendo um conjunto de frases variáveis por locutor e 3 conjuntos de frases comuns a todos os locutores, foneticamente balanceadas.

Foram usadas 4 arquiteturas de SOM com retícula quadrada para cada conjunto de treinamento: com 16, 25, 36 e 64 unidades de processamento. Cada combinação de uma arquitetura com um conjunto de frases foi treinada e testada. Um dos conjuntos de treinamento de frases comuns, foneticamente balanceadas, não foi usado para treinamento, sendo deixado como conjunto de referência para testes.

Os resultados dos testes indicaram que as arquiteturas com mais unidades possuem maior poder discriminativo, respondendo com menores erros de quantização durante o treinamento e maiores taxas de acerto durante os testes. Também indicaram que conjuntos de treinamento de maior duração favorecem o aumento das taxas de acerto.

O melhor desempenho em termos de taxas de acerto sobre o conjunto de referência foi conseguido com a arquitetura de 64 unidades, treinada com os conjuntos de frases foneticamente balanceadas, comuns aos locutores: 99.29%. Entretanto, esta mesma arquitetura apresentou taxas de acerto baixas quando testada no conjunto de frases va-

riáveis: 86.90%. Com o detalhamento deste erro por frase, observou-se que as frases de teste mais curtas foram responsáveis pelos erros, com taxas de acerto de 100% para as frases de teste mais longas. Isto definiu um limite inferior para a duração das frases de teste de aproximadamente 2.6s.

De forma geral, pode-se dizer que o conjunto, treinado com aproximadamente 17.5s de amostras de voz por locutor, é capaz de identificá-los com mais de 99% de taxa de acerto quando testado com locuções de duração superior a 2.8s, em modo independente de texto, colocando-o muito próximo aos sistemas estado da arte na categoria. Os resultados dos testes indicam que este desempenho pode ser ainda melhorado pelo aumento do número de unidades das SOMs.

5.2 Sobre o uso de uma SOM por locutor

A solução abordada apresenta uma característica arquitetural bastante interessante: para que um novo locutor seja acrescentado ao conjunto de locutores alvo, basta que uma nova SOM seja treinada para este locutor, sem necessidade de re-executar o processo de treinamento das demais. Da mesma forma, para que um locutor seja removido do conjunto alvo, basta destruir a sua SOM associada. Este desacoplamento dá maior viabilidade para aplicações em que o conjunto exato de locutores alvo é desconhecido *a priori*, desenvolvendo-se ao longo do ciclo de vida da aplicação.

Entretanto, o procedimento de identificação (teste) torna-se mais complexo com o aumento do número de locutores alvo: o número de redes em que um padrão deve ser testado aumenta, aumentando linearmente o tempo de resposta dos testes.

Outra questão que surge com o aumento do número de locutores é que as diferenças entre os erros totais de quantização entre as redes tende a ser mais próximo, refletindo a maior probabilidade de se encontrar locutores com vozes semelhantes. Entretanto, segundo os gráficos das figuras 25 e 26, o aumento do número de unidades das SOMs reduz tanto os erros de quantização durante o treinamento quanto as taxas de erro de identificação, conferindo às redes maior poder discriminativo.

Portanto, o número de unidades de processamento das SOMs deve ser baseado numa estimativa do número máximo de locutores que o conjunto deverá suportar, mantendo os níveis de desempenho aceitáveis. Além disto, deve-se levar em consideração que aplicações com um maior número de unidades de processamento por SOM tornam mais lentos os processos de treinamento e verificação, aumentando os tempos de resposta da aplicação.

5.3 Direções Futuras para Pesquisa

Neste trabalho, os índices das classes resultantes da quantização vetorial pelas SOMs foi ignorado: somente os erros totais de quantização foram utilizados. Testes preliminares, feitos fora do escopo desta dissertação, indicam que a seqüência destes índices de classe pode ser usada para identificar a frase sendo falada pelo locutor. O procedimento usado nestes testes consistiu na construção de mapas de transição de estado (histogramas de ocorrência de transições entre índices de classes), seguido da identificação através de redes neurais de arquitetura *Learning Vector Quantization* (LVQ). Experimentos anteriores com o uso deste método, aplicado na identificação de frequências de ondas mecânicas, propagadas numa sonda de prospecção de petróleo (SIMÕES, 2000), apresentaram resultados muito bons.

A solução apresentada deve ser testada sobre conjuntos com mais locutores, como por exemplo os *corpora* KING, SPOLTECH e TIMIT, buscando definir um compromisso entre o desempenho e a complexidade computacional da arquitetura, comparando este desempenho com os padrões atuais para estes conjuntos de amostras de voz. Também deve ser testada em conjuntos de vozes com diferentes níveis de ruído para que sua robustez seja avaliada.

Uma nova aplicação sugerida para este sistema é a segmentação de conversas entre um grupo de locutores: definir os trechos em que cada locutor está falando, os trechos de mistura de vozes e os trechos de silêncio. Esta segmentação é bastante interessante como pré-processamento para sistemas de reconhecimento de voz.

REFERÊNCIAS BIBLIOGRÁFICAS

ALCAIM, A.; SOLEWICZ, J. A.; MORAES, J. A. Frequência de ocorrência dos fones e listas de frases foneticamente balanceadas no Português falado no Rio de Janeiro. **Revista da Sociedade Brasileira de Telecomunicações**, v. 7, n. 1, p. 23–41, dez. 1992.

ANDERSON, T. R.; PATTERSON, R. Speaker recognition with the Auditory Image model and Self Organizing Feature Map: a comparison with traditional techniques. In: ESCA WORKSHOP ON SPEAKER RECOGNITION, IDENTIFICATION, AND VERIFICATION. **Proceedings**. Martigny, Switzerland: [s.n.], 1994.

ARBIB, M. A. (Ed.). **The Handbook of Brain Theory and Neural Networks**. Cambridge, MA: The MIT Press, 1995. 1118 p.

BECHETTI, C.; RICOTTI, L. P. **Speech Recognition: Theory and C++ implementation**. West Sussex, England: John Wiley & Sons, 1999. 407 p. Acompanha CD-ROM.

BENNANI, Y.; GALLINARI, P. A connectionist approach for speaker identification. In: ICASSP, 1990. **Proceedings**. Albuquerque, NM, USA: [s.n.], 1990. p. 265–268.

BENNANI, Y.; GALLINARI, P. On the use of TDNN-extracted features information in talker identification. In: ICASSP, 1991. **Proceedings**. Toronto, Ontario, Canada: [s.n.], 1991. p. 385–388.

BIMBOT, F.; MAGRIN-CHAGNOLLEAU, I.; MATHAN, L. Second-order statistical measures for text-independent speaker identification. **Speech Communication**, v. 17, n. 1–2, p. 177–192, ago. 1995.

BISHOP, C. M. **Neural Networks for Pattern Recognition**. Oxford: Oxford University Press, 1995. 482 p.

BOCCHIERI, E. L.; WILPON, J. G. Discriminative feature selection for speech recognition. **Computer Speech and Language**, v. 7, p. 229–246, 1993.

BORDEN, G.; HARRIS, K. **Speech Science Primer**: Physiology, acoustics, and perception. Baltimore, Md.: Williams & Wilkins, 1980.

CAMPBELL JR., J. P. Speaker recognition: a tutorial. **Proceedings of the IEEE**, v. 85, n. 9, p. 1437–1462, set. 1997.

COOLEY, J. W.; TUKEY, J. W. An algorithm for the machine computation of the complex Fourier series. **Mathematical Computation**, v. 19, p. 297–301, 1965.

DELACRETAZ, D. P.; HENNEBERT, J. Text-prompted speaker verification experiments with phoneme specific MLPs. In: ICASSP, 1998. **Proceedings**. [S.l.: s.n.], 1998. p. 777–780.

DELLER JR., J. R.; HANSEN, J. H. L.; PROAKIS, J. G. **Discrete-Time Processing of Speech Signals**. New York: Macmillan, 1993. 908 p.

DE MORI, R. (Ed.). **Spoken Dialogues with Computers**. London: Academic Publishers, 1998. 702 p.

DENES, P. B.; PINSON, E. N. **The Speech Chain**: The physics and biology of spoken language. 2 ed. New York: W. H. Freeman, 1993. 246 p.

DOYA, K. Recurrent networks: Supervised learning. In: ARBIB, M. A. (Ed.). **The Handbook of Brain Theory and Neural Networks**. Cambridge, MA: MIT Press, 1995. p. 796–800.

FARRELL, K. Networks for speaker recognition. In: KATAGIRI, S. (Ed.). **Handbook of Neural Networks for Speech Processing**. Norwood: Artech House Publishers, 2000. cap. 10, p. 357–391.

FARRELL, K.; MAMMONE, R. J.; ASSALEH, K. T. Speaker recognition using neural networks and conventional classifiers. **Acoustic, Speech and Signal Processing, IEEE Transactions on**, v. 2, n. 1, p. 194–205, jan. 1994. Part II.

FLANAGAN, J. L. **Speech Analysis, Synthesis and Perception**. 2 ed. New York: Springer-Verlag, 1972.

FURUI, S. An overview of speaker recognition technology. In: ESCA WORKSHOP ON SPEAKER RECOGNITION, IDENTIFICATION, AND VERIFICATION. **Proceedings**. Martigny, Switzerland: [s.n.], 1994.

FURUI, S. Recent advances in speaker recognition. **Pattern Recognition Letters**, v. 18, n. 9, p. 859–872, set. 1997.

FURUI, S. Speaker recognition. In: Ron Cole; Victor Zue (Ed.). **Survey of the State of Art in Human Language Technology**. [S.l.]: National Science Foundation and European Union, 1996. cap. 1.7, p. 42–48. Disponível em: [<http://cslu.cse.ogi.edu/HLTsurvey/>](http://cslu.cse.ogi.edu/HLTsurvey/).

GERSHO, A.; GRAY, R. M. **Vector Quantization and Signal Compression**. Norwell, MA: Kluwer Academic Publishers, 1992. (The Kluwer International Series in Engineering and Computer Science).

GISH, H.; SCHMIDT, M. Text-independent speaker verification. **IEEE Signal Processing Magazine**, v. 11, n. 4, p. 18–32, out. 1994.

HAYKIN, S. **Neural Networks: A comprehensive foundation**. 2 ed. Englewood Cliffs, NJ: Prentice Hall, 1999. 842 p.

HOMAYOUNPOUR, M. M.; CHOLLET, G. Neural Net approaches to speaker verification: comparison with second order statistic measure. In: ICASSP, 1995. **Proceedings**. Detroit, MI, USA: [s.n.], 1995. v. 1, p. 353–356.

INTRATOR, N. Competitive learning. In: ARBIB, M. A. (Ed.). **The Handbook of Brain Theory and Neural Networks**. Cambridge, MA: MIT Press, 1995. p. 220–222.

JUANG, B. H.; RABINER, L. R.; WILPON, J. G. On the use of bandpass liftering in speech recognition. **Acoustic, Speech and Signal Processing, IEEE Transactions on**, v. 35, p. 947–954, jul. 1987.

KATAGIRI, S. (Ed.). **Handbook of Neural Networks for Speech Processing**. Norwood: Artech House Publishers, 2000. 522 p.

KOENIG, W. A new frequency scale for acoustic measurements. **Bell Telephone Laboratory Record**, v. 27, p. 299–301, 1949.

KOHONEN, T. **Learning vector quantization for pattern recognition**. Helsinki: Helsinki University of Technology, 1986. (Technical Report, TTK-F-A601).

KOHONEN, T. The self-organizing map. **Proceedings of the IEEE**, v. 78, n. 9, p. 1464–1480, set. 1990.

KOHONEN, T. **Self-Organizing Maps**. 3 ed. Berlin: Springer-Verlag, 2001. (Springer Series in Information Sciences, 30).

KUMAGAI, J. Talk to the machine. **IEEE Spectrum**, p. 60–64, set. 2002. Special R&D Report.

LANG, K. J.; HINTON, G. E. **The development of the time-delay neural network architecture for speech recognition**. Carnegie-Mellon University: Pittsburg, PA, 1988. (Technical Report, TR-CMU-CS-88-152).

LAPIDOT, I.; GUTERMAN, H.; COHEN, A. Unsupervised speaker recognition based on competition between Self-Organizing Maps. **Neural Networks, IEEE Transactions on**, v. 13, n. 4, p. 877–887, jul. 2002.

LECUN, Y. Une procedure d'apprentissage pour reseau a seuil assymetrique. **Cognitiva**, v. 85, p. 599–604, 1985.

LEE, C. H.; SOONG, F. K.; PALIWAL, K. K. (Ed.). **Automatic Speech and Speaker Recognition: Advanced topics**. Boston: Kluwer Academic Publishers, 1996. 517 p.

MATSUI, T.; FURUI, S. Speaker adaption of tied-mixture-based phoneme models for text-prompted speaker recognition. In: **Proceedings**. [S.l.: s.n.], 1994. v. 1, p. I/125–I/128.

MERMELSTEIN, P.; DAVIS, B. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. **Acoustic, Speech and Signal Processing, IEEE Transactions on**, v. 28, n. 4, p. 357–366, ago. 1980.

MINSKY, M. L.; PAPERT, S. A. **Perceptrons: Expanded edition**. 2 ed. Cambridge, MA: The MIT Press, 1990. 292 p.

NAIK, J. M.; NETCH, L. P.; DODDINGTON, G. R. Speaker verification over long distance telephone lines. In: ICASSP, 1989. **Proceedings**. Glasgow, Scotland: [s.n.], 1989. p. 524–527.

OGLESBY, J.; MASON, J. S. Optimization of neural models for speaker identification. In: ICASSP, 1990. **Proceedings**. Albuquerque, NM, USA: [s.n.], 1990. p. 261–264.

OPPENHEIM, A. V.; SCHAFER, R. W. **Discrete-Time Signal Processing**. 2 ed. Upper Saddle River, NJ: Prentice Hall, 1998. 870 p. (Prentice Hall Signal Processing Series).

PARKER, D. B. **Learning-logic: casting the cortex of the human brain in silicon**. Cambridge, MA: Center for Computational Research in Economics and Management Science, 1985. (Technical Report, TR-47).

PHILLIPS, P. J. et al. An introduction evaluating biometric systems. **Computer Magazine**, v. 33, n. 2, p. 56–63, fev. 2000.

PICONE, J. W. Signal modeling techniques in speech recognition. **Proceedings of the IEEE**, v. 81, n. 9, p. 1215–1247, set. 1993.

PRESS, W. H. et al. **Numerical Recipes in C: The art of scientific computing**. 2 ed. Cambridge, UK: Cambridge University Press, 1992. 994 p.

RABINER, L. R.; JUANG, B. W. **Fundamentals of Speech Recognition**. Englewood Cliffs, NJ: Prentice Hall, 1993. 507 p.

RABINER, L. R.; WILPON, J. G.; SOONG, F. K. High performance connected digit recognition using hidden Markov models. **Acoustic, Speech and Signal Processing, IEEE Transactions on**, v. 37, p. 1214–1225, ago. 1989.

RABINER, L. A tutorial on Hidden Markov Models and selected applications in speech recognition. **Proceedings of the IEEE**, v. 77, n. 2, p. 257–286, fev. 1989.

REYNOLDS, D. A.; QUATIERI, T. F.; DUNN, R. B. Speaker verification using adapted Gaussian Mixture Models. **Digital Signal Processing Review Journal**, jan. 2000.

REYNOLDS, D. Speaker identification and verification using Gaussian Mixture Models. **Speech Communications**, v. 17, p. 91–108, ago. 1995.

REYNOLDS, D. A.; HECK, L. P. Automatic speaker recognition: recent progress, current applications, and future trends. In: AMERICAN ASSOCIATION FOR THE ADVANCEMENT OF SCIENCE (AAAS) SYMPOSIUM. Washington, DC: [s.n.], 2000.

ROSENBERG, A. E.; LEE, C. H.; GOKEEN, S. Connected word talker recognition using whole word Hidden Markov Models. In: ICASSP, 1991. **Proceedings**. Toronto, Ontario, Canada: [s.n.], 1991. p. 381–384.

ROSENBERG, A. E.; LEE, C. H.; SOONG, F. K. Sub-word unit talker verification using Hidden Markov Models. In: ICASSP, 1990. **Proceedings**. Albuquerque, NM, USA: [s.n.], 1990. p. 269–272.

ROSENBLATT, F. The Perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**, v. 65, p. 386–408, 1958.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **Nature**, London, v. 323, p. 533–536, 1986.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning internal representations by error propagation. In: RUMELHART, D. E.; MCCLELLAND, J. L. (Ed.). **Parallel Distributed Processing: Explorations in the microstructure of cognition**. Cambridge, MA: MIT Press, 1986, v. 1: Foundations, v. 1: Foundations. cap. 8, p. 318–362.

RUMELHART, D. E.; ZIPSER, D. Feature discovery by competitive learning. In: RUMELHART, D. E.; MCCLELLAND, J. L. (Ed.). **Parallel Distributed Processing: Explorations in the microstructure of cognition**. Cambridge, MA: MIT Press, 1986, v. 1: Foundations, v. 1: Foundations. cap. 5, p. 151–193.

SANKAR, A.; MAMMONE, R. J. Growing and pruning neural tree networks. **Computers, IEEE Transactions on**, v. C-42, p. 221–229, mar. 1993.

SHIKANO, K. **Evaluation of LPC spectral matching measures for phonetic unit recognition**. New York: Computer Science Department, Carnegie-Mellon University, maio 1985. (Technical Report).

SIMÕES, M. G. et al. A novel competitive learning neural network based acoustic transmission system for oil-well monitoring. **Industry Applications, IEEE Transactions on**, v. 36, n. 2, p. 484–491, abr. 2000.

SOONG, F. K. et al. A vector quantization approach to speaker recognition. In: ICASSP, 1985. **Proceedings**. [S.l.: s.n.], 1985. p. 387–390.

SÓRIA, R. A. B. **Reconhecimento Automático de Locutor usando pré-processamento em sons nasalizados com diversos classificadores neurais.**

Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo, São Paulo, SP, 2001.

SRINIVASAN, S.; BROWN, E. Is speech recognition becoming mainstream?

Computer Magazine, abr. 2002. Disponível em: <<http://computer.org/computer/homepage/0402/GEI/index.htm>>. Acesso em: 26 .apr. 2002.

STEVENS, S. S.; VOLKMAN, J. The relation of pitch to frequency. **American Journal of Psychology**, v. 53, p. 329, 1940.

TEXAS INSTRUMENTS INC. **Understanding Data Converters**. [S.l.], 1995. (Application Report, SLAA013).

TSOI, A. C. et al. Application of artificial neural network techniques to speaker verification. In: ESCA WORKSHOP ON SPEAKER RECOGNITION, IDENTIFICATION, AND VERIFICATION. **Proceedings**. Martigny, Switzerland: [s.n.], 1994.

MALSBURG, C. von der. Self-organization of orientation sensitivity cells in the striate cortex. **Kybernetik**, n. 14, p. 85–100, 1973.

WAIBEL, A. et al. Phoneme recognition using time-delay neural networks. **Acoustic, Speech and Signal Processing, IEEE Transactions on**, v. ASSP-37, n. 3, p. 328–339, 1989.

WERBOS, P. J. **Beyond regression: new tools for prediction and analysis in the behavioral sciences**. Tese (Doutorado) — Harvard University, Cambridge, MA, 1974.

YUK, D. S. et al. **A neural network system for robust large-vocabulary continuous speech recognition in variable acoustic environments**. Piscataway, N.J.: Rutgers - The State University of New Jersey, 15 jan. 1999. (Technical Report CAIP-TR-234).