# COMPRESSÃO DE SINAIS DE FALA UTILIZANDO REDES NEURAIS

Speech Signal Compression Using Neural Networks

**Mário ULIANI NETO**
Faculdade de Jaguariúna
Faculdade Politécnica de Campinas
Fundação CPqD

**Flávio Olmos SIMÕES**
Fundação CPqD

**Jeremias Barbosa MACHADO**
Universidade Estadual de Campinas - UNICAMP

**Resumo:** Propomos aqui uma técnica de compressão de fala baseada em quantização vetorial. Uma rede neural com treinamento não supervisionado é usada para implementar o quantizador. Idéias gerais do problema de quantização vetorial são abordadas em conjunto com aspectos introdutórios relativos ao processamento de sinais de fala. Em seguida, é mostrado como a técnica de quantização vetorial pode ser empregada para construir um codebook de fala. As redes de Kohonen bidimensionais são apresentadas como uma ferramenta para a geração do codebook. Finalmente, são apresentados resultados de simulação evidenciando as melhores estratégias de inicialização e treinamento da rede, assim como a melhor topologia da rede para o problema em questão.
**Palavras-chave:** Compressão de sinais de fala; quantização vetorial; redes neurais com treinamento não supervisionado; processamento de sinais.

**Abstract:** We propose a speech compression technique based on vector quantization. A neural network with unsupervised learning is used to implement the vector quantizer. Some general issues concerning the vector quantization problem are presented, as well as some basic aspects related to speech signal processing. The idea of using a codebook to perform speech compression is introduced, and the use of a 2-dimensional self-organizing Kohonen map to generate the codebook is proposed. Finally, simulation results are presented, giving some insights on the best network initialization and training strategies, as well as the best network topology for this problem.
**Key-words:** Speech compression. Vector quantization. Neural networks with unsupervised learning. Signal processing.

## Introdução

Uma base de dados cujos elementos são vetores de dimensão fixa pode ser armazenada de forma bastante compacta através da utilização da técnica conhecida como quantização vetorial. Nesta técnica, os vetores da base de dados original são

substituídos por vetores aproximados, extraídos de um inventário gerado previamente, denominado codebook. A eficácia do processo de compressão via quantização vetorial pode ser medida a partir do erro de quantização introduzido na nova representação dos dados. No projeto do codebook, visa-se à minimização do erro de quantização médio.

Quadros de sinais de fala parametrizados são um exemplo de dados que podem ser armazenados em notação vetorial e, portanto, podem ser submetidos a um processo de compressão via quantização vetorial.

O objetivo deste trabalho é apresentar uma estratégia de compressão de sinais de fala baseada na divisão do sinal em quadros síncronos com o período de pitch, seguida da representação dos quadros como vetores de parâmetros e da compressão do conjunto de quadros via quantização vetorial.

Na proposta aqui apresentada, utiliza-se uma rede neural bidimensional do tipo SOM (self-organizing map) (RUNSTEIN, 1998) em conjunto com o algoritmo de clusterização k-means para implementar o processo de clusterização dos quadros da base de treinamento, a fim de gerar o codebook de fala. O codebook gerado é usado na compressão de novos sinais de fala, através da transformação da seqüência de quadros do sinal a ser codificado em uma seqüência de índices do codebook. No processo de decodificação, a seqüência de índices do codebook é transformada novamente em uma seqüência de quadros e os quadros recuperados são utilizados na reconstrução do sinal através da aplicação da técnica de *overlap and add*.

Na Seção 1 deste trabalho é apresentada uma breve introdução ao problema de quantização vetorial. Na Seção 2 são apresentados alguns conceitos relacionados à representação e ao processamento de sinais de fala. Na Seção 3 mostra-se como a técnica de quantização vetorial pode ser aplicada à compressão de sinais de fala. A Seção 4 apresenta uma discussão sobre redes neurais auto-organizáveis e sua utilização na clusterização de dados. Por fim, na Seção 5 são apresentados resultados da aplicação de uma rede auto-organizável na quantização vetorial de uma base de dados composta por quadros parametrizados extraídos de sinais de fala. Dentre as alternativas estudadas, é feita uma discussão indicando

quais se apresentaram como o melhor conjunto de parâmetros a ser usado para representar os quadros, a melhor topologia de rede e as melhores estratégias de inicialização e treinamento da rede neural.

## 1. Quantização Vetorial e Compressão de Dados

Quantização vetorial é uma técnica clássica de compressão de dados bastante utilizada em aplicações como compressão de imagens, compressão de voz e reconhecimento de fala, dentre outras (GERSHO & GRAY, 1992; GRAY & NEUHOFF, 1998). Trata-se de uma técnica de compressão que normalmente acarreta perdas, uma vez que um codebook de tamanho limitado pode não permitir a recuperação na íntegra da informação original (HAYKIN, 1999).

Em quantização vetorial, tem-se um conjunto de dados de entrada $\{v_1, v_2, \ldots, v_N\}$ constituído por $N$ vetores de dimensão $k$. A tarefa de um compressor baseado em quantização vetorial é fazer o mapeamento desse conjunto de entrada em um outro conjunto $\{c_1, c_2, \ldots, c_M\}$, de tamanho finito $M \square N$, também composto por vetores de dimensão $k$. Esse novo conjunto é denominado codebook e os vetores $c_i$ que o compõem são denominados codevectors.

No processo de compressão, cada um dos vetores do conjunto de entrada será mapeado em um codevector. Daí advém o nome da técnica, pois o novo vetor será uma versão quantizada (ou seja, aproximada) do vetor original. Esta idéia é ilustrada na Figura 1.

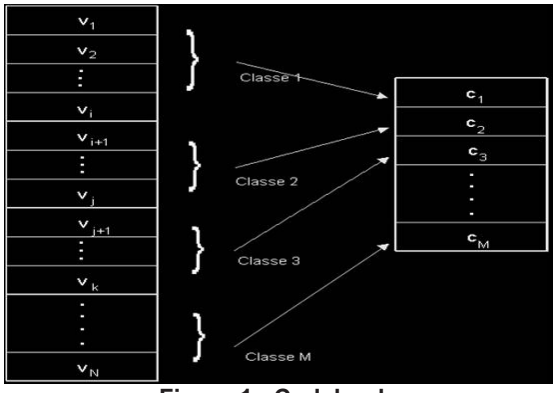


**Figura 1 - Codebook.**

Esta aproximação introduz um erro na representação dos dados de entrada, denominado erro de quantização vetorial (GRAY & NEUHOFF, 1998). Supondo que haja uma métrica $d(\cdot,\cdot)$ que represente a distância entre dois vetores (por exemplo, a distância euclidiana), uma medida do erro de quantização de um vetor $v_i$ do conjunto original é a distância entre esse vetor $v_i$ e a sua versão quantizada $q(v_i)$:

$$Q_i = d(v_i, q(v_i)) \qquad \textbf{(1)}$$

No projeto do codebook, deseja-se minimizar a perda de informação do conjunto de dados codificados. Em outras palavras, para cada vetor de entrada $v_i$, deseja-se minimizar o erro de quantização dado por (1). Minimizar o erro de quantização para todos os vetores do conjunto de entrada equivale a minimizar o erro de quantização vetorial médio do conjunto, o qual pode ser definido por

$$Q_N = \frac{1}{N}\sum_{i=1}^{N} Q_i \qquad \textbf{(2)}$$

onde N representa o número de vetores do conjunto de entrada.

Para criar um codebook que atenda esse critério, é necessário escolher um conjunto de codevectors de forma que as distâncias entre codevectors e vetores por eles representados sejam as menores possíveis (GRAY & NEUHOFF, 1998).

No processo de geração do codebook, utiliza-se um conjunto de dados de treinamento representativo dos dados a serem posteriormente codificados. Os dados de treinamento são agrupados em classes ou clusters, de acordo com algum critério de proximidade (normalmente, a mesma métrica de distância usada no cálculo do erro de quantização). Um cluster é, portanto, um subconjunto de vetores suficientemente próximos entre si. O número de clusters a ser gerado depende da distribuição dos dados de treinamento. Normalmente, deseja-se minimizar a distância intragrupo e maximizar a distância intergrupo. É possível, mas não obrigatório, fixar a priori o número de clusters, com o objetivo de gerar um codebook de tamanho previamente conhecido. Tal estratégia, no entanto, não garante a obtenção do menor erro de quantização médio. Em princípio, quanto maior o número de clusters, menor o erro de quantização médio.

Uma vez gerado o codebook, este é utilizado para compressão dos dados da seguinte forma: para cada vetor do conjunto de entrada, varre-se o codebook em busca do codevector mais próximo, associando-se ao vetor de entrada o índice relativo a este codevector. Desta forma, o conjunto de vetores de entrada é transformado em um conjunto de índices do codebook. Em uma aplicação em que haja transmissão de dados, apenas os índices são transmitidos pelo canal e não os codevectors. Supõe-se portanto que o codebook seja conhecido também no receptor.

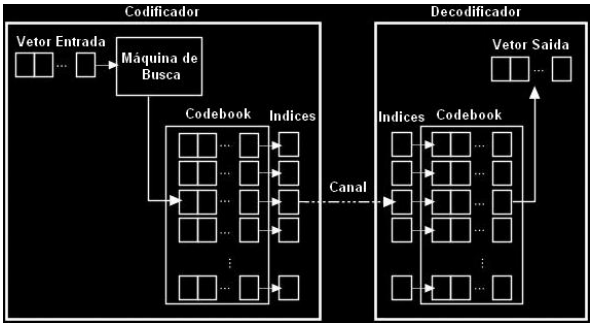


**Figura 2 - Codificação e decodificação por meio de codebook.**

Na etapa de decodificação, os índices são recebidos e usados na recuperação da seqüência de codevectors. O processo completo é ilustrado na Figura 2.

## 2. Análise de Sinais de Fala

Um sinal de fala é produzido a partir da passagem do ar pelo aparelho fonador humano. Ao ser representado digitalmente, o sinal de fala pode ser visto como um conjunto de amostras espaçadas no eixo do tempo.

As características do sinal de fala em um dado instante dependem da configuração momentânea do trato vocal do falante, ou seja, da abertura dos lábios e da mandíbula, da posição da língua, da taxa de vibração das pregas vocais, etc. Ao proferir uma sentença, o falante modifica continuamente a configuração de seu trato vocal, de forma a produzir uma seqüência de sons que transmite uma mensagem ao ouvinte. Essa seqüência de sons é composta por unidades básicas denominadas fones. Pode-se definir um fone como um trecho do sinal de fala cujas características acústicas seguem um determinado padrão.

Nos trechos de sinal de fala vozeados (Figura 3), ocorre a vibração das pregas vocais. Percebe-se nesse caso que o sinal de fala apresenta uma característica quase periódica, com a ocorrência de picos que se repetem com espaçamento aproximadamente constante. O espaçamento entre esses picos está diretamente relacionado à taxa de vibração das pregas vocais: picos mais próximos

entre si indicam uma maior taxa de vibração e, por conseqüência, uma voz mais aguda; picos mais espaçados, por sua vez, indicam uma taxa de vibração menor das pregas vocais e uma voz mais grave.

Já nos trechos não-vozeados do sinal (Figura 4), não ocorre vibração das pregas vocais. Nesse caso, o sinal possui energia mais baixa do que a dos trechos vozeados e apresenta característica totalmente aperiódica, assemelhando-se a um sinal de ruído.

Há, por fim, trechos do sinal de fala com características híbridas entre as dos sinais vozeados e as dos não-vozeados.

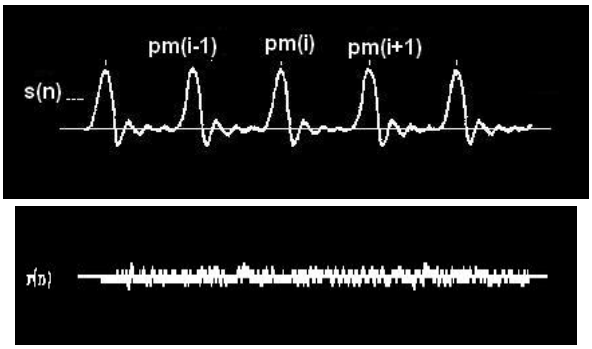


**Figura 4 - Sinal não-vozeado.**

Muito embora as características do sinal de fala variem continuamente ao longo do tempo, é possível analisar as suas características acústicas de forma discreta. Uma maneira de fazer isso é subdividir o sinal de fala em trechos de curta duração, chamados quadros, nos quais as características acústicas podem ser consideradas praticamente constantes.

A divisão do sinal de fala em quadros adotada neste trabalho diferencia trechos de sinal vozeados dos não-vozeados, através do uso de marcas de pitch, conforme definido a seguir. Nos trechos do sinal com característica vozeada, as marcas de pitch são posicionadas nos picos do sinal de fala (Figura 3). A distância entre marcas de pitch consecutivas, denominada período de pitch, está relacionada com com a taxa de vibração das pregas vocais naquele trecho do sinal. Nos sinais

não-vozeados, as marcas de pitch são posicionadas em instantes igualmente espaçados no tempo e não têm relação com a vibração das pregas vocais.

Uma vez posicionadas as marcas de pitch no sinal de fala, faz-se a subdivisão desse sinal em quadros. Neste trabalho, um quadro é definido como sendo um trecho de sinal em torno de uma marca de pitch, iniciando na marca de pitch imediatamente anterior e indo até a marca de pitch imediatamente seguinte. Percebe-se, portanto, que há sobreposição entre quadros adjacentes: as amostras do período direito de um quadro são as mesmas do período esquerdo do quadro seguinte.

Para extrair um quadro do sinal de fala, multiplica-se o trecho de fala de interesse por uma janela assimétrica cujo pico coincide com a marca de pitch central do quadro e com o mesmo número de amostras do quadro. Trata-se de uma janela formada pela junção de duas meias janelas de Hanning (MAKHOUL & WOLF, 1972; BOLL, 1979). A primeira metade corresponde à metade esquerda de uma janela de Hanning de tamanho $2N_1$ , e a segunda metade corresponde à metade direita de uma janela de Hanning de tamanho $2N_2$ . As amostras da janela, com primeira amostra na origem são dadas pela seguinte expressão:

$$w(n)=\begin{cases}0{,}5\left[1-\cos\left(\dfrac{\pi n}{N_1-1}\right)\right]; & 0\le n\le N_1\\ 0{,}5\left[1-\cos\left(\dfrac{\pi\left(n+N_1-N_2+1\right)}{N_2}\right)\right]; & N_1\le n\le N_2\end{cases} \quad \textbf{(3)}$$

Ao multiplicar-se o sinal de fala por uma janela posicionada na marca de pitch central do quadro sob análise, obtém-se o quadro janelado, cujas amostras correspondem às amostras originais do quadro com atenuação crescente em direção às bordas.

No processo de janelamento de quadros adjacentes, as janelas são posicionadas de forma que a primeira amostra de uma janela coincida com a amostra da marca central da janela anterior e a última amostra dessa mesma janela coincida com a amostra da marca central da janela seguinte. Ao somar as amostras

de janelas consecutivas posicionadas dessa maneira, em um processo chamado *overlap and add*, obtém-se uma seqüência de amostras de valor constante igual a 1.

Essa propriedade permite reconstruir o sinal original sem distorção a partir de seus quadros janelados. Para isso, basta posicionar os quadros janelados conforme descrito anteriormente e em seguida fazer o overlap and add dos mesmos. A Figura 5 ilustra esse processo.

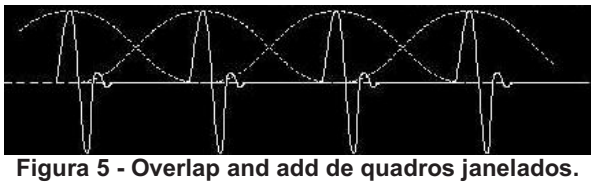

**Figura 5 - Overlap and add de quadros janelados.**

## 3. Quantização Vetorial Aplicada à Compressão de Sinais de Fala

Quadros de um sinal de fala correspondentes a sons similares podem apresentar forte semelhança entre si. É possível explorar essa semelhança a fim de armazenar sinais de fala de forma mais compacta. A idéia por trás dessa estratégia é descartar informações repetitivas presentes em quadros similares, armazenando somente as diferenças relevantes. Uma maneira de implementar essa estratégia é através de quantização vetorial (BUZO et al., 1981; KRISHNAMURTHY et al., 1990).

Para que a quantização vetorial possa ser usada na compressão de sinais de fala, é necessário que o sinal seja representado como um conjunto de vetores de dimensão fixa. Para que possamos utilizar os quadros janelados como sendo as unidades básicas que serão sujeitas ao processo de quantização vetorial, devemos realizar uma transformação na forma de representar os quadros. Isso porque o número de amostras de um quadro é variável: quanto maior o período de pitch associado ao quadro, maior o número de amostras nele presentes.

Essa transformação é denominada parametrização. Cada quadro passará a ser representado por um conjunto fixo de parâmetros associados a características acústicas do sinal de fala. Quanto mais semelhantes forem os quadros em termos acústicos, mais próximos devem ser os seus vetores de parâmetros.

Para implementar um processo de quantização vetorial de forma eficiente, é necessário definir um conjunto de parâmetros que carregue a maior quantidade possível de informação relevante para diferenciação entre quadros. É importante também que haja baixa correlação entre os parâmetros usados, a fim de evitar que estes carreguem informação redundante.

Uma vez definido o conjunto de parâmetros a ser utilizado, cada quadro do conjunto de treinamento é associado a um vetor de parâmetros. Esses vetores são agrupados em clusters e, para cada cluster, elege-se um vetor representante, que será o codevector a ser incluído no codebook. O codevector será escolhido dentre os vetores que compõem o cluster, de forma a minimizar o erro de quantização médio dos vetores do cluster.

Diferentes técnicas podem ser consideradas para gerar o codebook. A Seção 4 apresenta a estratégia implementada neste trabalho, baseada em redes neurais com treinamento não-supervisionado (mapa de Kohonen bidimensional).

Uma vez construído o codebook, é possível utilizá-lo para codificar um sinal de fala qualquer. O processo de codificação consiste dos seguintes passos:

- divisão do sinal a ser codificado em quadros;
- determinação dos períodos de pitch esquerdo e direito associados aos quadros;
- determinação da energia dos quadros;
- geração dos vetores de parâmetros dos quadros;
- mapeamento dos vetores do quadros em índices do codebook.

Para cada quadro do sinal codificado, são transmitidos ao receptor o índice do codevector, a energia e os períodos esquerdo e direito (Figura 6). No decodificador, a seqüência de índices é mapeada novamente em uma seqüência de codevectors (Figura 7).

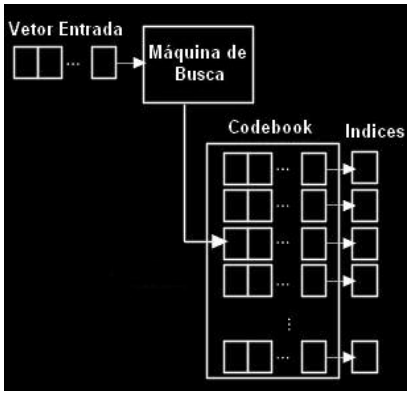


**Figura 6 - Codificação de quadros de fala usando codebook.**

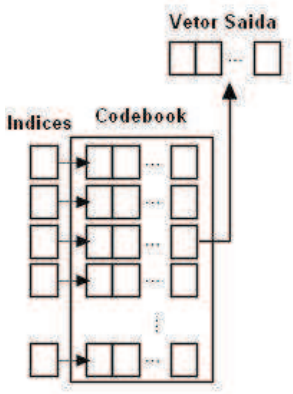


**Figura 7 - Decodificação de quadros de fala usando codebook.**

Não é possível reconstruir as amostras dos quadros janelados correspondentes aos codevectors apenas a partir dos parâmetros acústicos armazenados no vetor. Por isso, é necessário que haja no decodificador um dicionário de quadros, que armazene, para cada codevector, o quadro janelado que o gerou na etapa de treinamento. Dessa forma, pode-se transformar a seqüência de índices que chega ao receptor em uma seqüência de quadros janelados.

Os quadros janelados recuperados do dicionário são submetidos a um ajuste de ganho, de forma que a sua energia passe a ser igual à energia do quadro original. Com isso, evita-se a ocorrência de descontinuidades de amplitude no sinal reconstruído.

A reconstrução do sinal é feita através da operação de sobreposição dos quadros janelados (*overlap and add*) após a correção de ganho (Figura 8). O espaçamento entre as janelas deve ser tal que os períodos de pitch do sinal original sejam preservados. Dessa forma, a curva de entonação da sentença reconstruída será a mesma da sentença original, assim como a sua duração.

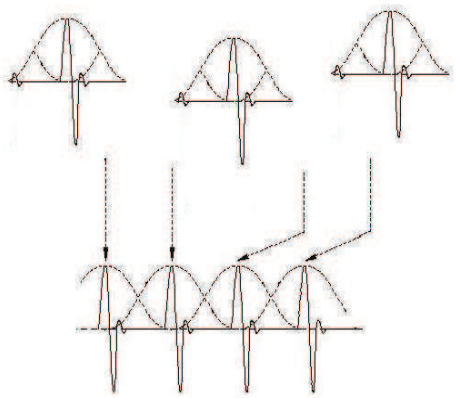

**Figura 8 - Reconstrução do sinal de fala utilizando overlap and add.**

Deve-se observar que, ao fazer este posicionamento das janelas, não existe mais garantia de que a primeira amostra de uma janela coincida com a amostra da marca central da janela anterior, nem de que a última amostra dessa mesma janela coincida com a amostra da marca central da janela seguinte. Isso porque o período do quadro armazenado no codebook não é necessariamente o mesmo do quadro sintetizado. Essa alteração no nível de sobreposição das janelas introduz distorção no sinal reconstruído. Tal distorção será tão menor quanto menor for a diferença entre os períodos de pitch dos quadros originais e os dos quadros recuperados do dicionário.

## 4. Mapas Auto Organizáveis

Os mapas auto-organizáveis (SOM – *self-organizing maps*) são um tipo de rede neural artificial inicialmente proposta por Kohonen (KOHONEN, 1982), levando seu nome. Seu treinamento é feito através de processo não-supervisionado que consiste de 4 etapas (HAYKIN, 1999):

- inicialização da rede;
- processo competitivo;
- processo cooperativo;
- processo adaptativo.

Na inicialização da rede são definidos os seus parâmetros de configuração: o tipo de distribuição dos neurônios (uni ou bidimensional); a topologia a ser utilizada,

no caso bidimensional (folha, cilíndrica ou toroidal) e os valores iniciais dos pesos sinápticos dos neurônios, que podem ser definidos aleatoriamente ou por algum outro critério.

Na etapa de aprendizagem propriamente dita, padrões de treinamento são apresentados seqüencialmente à rede. Nesta etapa, o vetor $x$, que representa um objeto de entrada, é apresentado a todos os neurônios da rede e é calculada a distância entre $x$ e todos os vetores de pesos dos respectivos neurônios da rede. A unidade do mapa que apresenta a menor distância em relação ao vetor de entrada é denominada neurônio vencedor ou BMU (*Best Matching Unit*). Essa etapa é classificada como a etapa de competição (KOHONEN, 1990a; KOHONEN, 1990b), na qual se deseja encontrar o neurônio com maior ativação para cada objeto de entrada.

Em seguida, o neurônio vencedor (correspondente ao BMU) é atualizado, sendo movido em direção ao padrão apresentado na entrada. Os neurônios presentes na vizinhança do neurônio vencedor são também atualizados na direção do padrão de entrada, com taxa de deslocamento seguindo geralmente uma distribuição gaussiana ao redor do neurônio vencedor. Dessa maneira, a vizinhança topológica decresce com o passar do tempo (HAYKIN, 1999). Esta etapa é conhecida como etapa cooperativa, uma vez que o neurônio vencedor irá influenciar no ajuste da sua vizinhança.

O processo de aprendizagem se conclui com a etapa adaptativa. Nesta etapa, os vetores de pesos sinápticos $w$ dos neurônios serão ajustados em direção ao vetor de entrada $x$. Os padrões de entrada são apresentados a rede até que se atinja um dado número de iterações ou que um determinado critério de parada seja satisfeito (por exemplo, quando o erro de quantização médio ficar abaixo de um dado limiar).

### 4.1. Matriz U

Os mapas auto-organizáveis são uma projeção de espaços de alta dimensão dos dados de entrada no espaço de dimensão reduzida da estrutura da rede de Kohonen, que normalmente é uni ou bi-dimensional. O algoritmo de aprendizagem de um SOM é projetado para preservar as relações de vizinhança do espaço de alta dimensão no espaço do mapa (KOHONEN, 1982).

Para se analisar as características emergentes do mapa de Kohonen, foi proposto o uso da matriz U (ULTSCH & SIEMON, 1990). A matriz U é construída a partir da superfície gerada pela plotagem das distâncias entre dois neurônios vizinhos e representa o comportamento emergente do mapa de Kohonen a partir dos objetos presentes no espaço de entrada.

### 4.2. Clusterização e o Algoritmo k-means

A matriz U representa a distância entre os pesos dos neurônios, preservando a vizinhança topológica dos padrões de entrada que geralmente estão em um espaço de alta dimensão. Muitas vezes é difícil determinar com precisão onde se localizam as fronteiras dos clusters. Em determinadas aplicações, a matriz U é suficiente para se determinar os clusters e suas fronteiras. Contudo, no problema abordado neste trabalho, além de determinar os clusters é necessário determinar o neurônio que melhor representa cada cluster, para que o mesmo esteja presente no codebook no processo de quantização vetorial.

Para solucionar este problema, é utilizado o algoritmo k-means de clusterização sobre a matriz U. O algoritmo k-means (FORGEY, 1965; MACQUEEN, 1967) permite criar clusters a partir de um conjunto de objetos, tendo como objetivo encontrar protótipos para os clusters de forma que a distância dos objetos dentro do grupo em relação a estes protótipos seja mínima. Um dos mais populares algoritmos utilizados para resolver de forma iterativa este problema é o algoritmo de Lloyd (LLOYD, 1982). Este algoritmo inicia-se com o particionamento, aleatório ou utilizando alguma heurística, dos objetos de entrada em clusters. Posteriormente, é calculado o protótipo de cada um dos clusters, dado inicialmente como o ponto médio. Uma vez calculado o protótipo, os clusters são redefinidos de forma a

minimizar a distância dos objetos ao centróide. Após a redefinição, novos centróides são calculados e os clusters são novamente redefinidos. O processo é repetido até que uma dada condição de convergência seja atingida.

Após a convergência, são definidos os clusters e seus respectivos protótipos, os quais serão utilizados como representantes dos clusters em aplicações do codebook obtido.

## 5. Análise de Resultados

Neste trabalho, a rede de Kohonen foi utilizada em conjunto com o algoritmo k-means para fazer o agrupamento dos quadros de fala de acordo com suas características acústicas. Para tal, diferentes quadros pertencentes a uma base de dados de fala foram apresentados à rede em lote. Pelo fato de que quadros de fala adjacentes no tempo muitas vezes apresentam alta correlação, tomou-se o cuidado de embaralhá-los a cada iteração antes da apresentação dos dados à rede. O treinamento em lote utilizou como critério de parada um número de iterações fixo e suficientemente grande, de forma a garantir que o erro de quantização atingisse um regime permanente (convergência). Como resultado final do treinamento, os neurônios da rede representam os diferentes grupos de quadros de fala apresentados à rede.

Os ensaios foram divididos em duas partes: a primeira delas consiste na análise dos parâmetros da fala, em busca do conjunto capaz de gerar os melhores resultados; a segunda consiste na análise topológica referente à inicialização e ao treinamento da rede neural empregada, buscando avaliar a influência dos diferentes arranjos na qualidade do sinal de fala sintetizado a partir do sinal quantizado. Por fim, foi feita a comparação dos resultados obtidos por meio da técnica proposta com os resultados de um algoritmo de codificação perceptual popular, de forma a obter uma referência do potencial do método aqui proposto. Nessa etapa de análise, utilizou-se a mesma base de dados de fala tanto para o treinamento do codebook como para validação do processo de compressão. A base de dados utilizada é composta por arquivos de fala representando 100 frases. As frases foram projetadas

de forma a apresentar riqueza e diversidade fonéticas. Os arquivos de fala são armazenados no formato wave (codificação PCM linear) com taxa de amostragem de 16 kHz e 16 bits por amostra. No total, as 100 frases da base de dados são constituídas por aproximadamente 52000 quadros.

### 5.1. Seleção de Parâmetros

O principal objetivo desta análise é determinar um conjunto de parâmetros que agrupe os quadros do sinal de fala de forma coerente. Para tal, foi definida uma metodologia capaz de avaliar de maneira objetiva e quantitativa a qualidade do sinal de fala reconstruído a partir dos quadros presentes no codebook. Essa metodologia consiste em gerar diferentes conjuntos de vetores de atributos a partir dos quadros da base de dados de fala. Os vetores são agrupados em clusters utilizando-se uma rede de Kohonen em conjunto com o algoritmo k-means para formar o codebook de fala. A partir do codebook, foram reconstruídas as mesmas 100 frases usadas no treinamento e a qualidade desses sinais foi avaliada. Portanto, o objetivo da metodologia apresentada é avaliar o conjunto de parâmetros que gerou o melhor codebook.

Para avaliação da qualidade de sinais de fala foram utilizadas técnicas de avaliação objetiva de sinais de voz (ITU-T P.861, 1998; ITU-T P.862, 2001). Tais técnicas utilizam modelos psicoacústicos que, com base em diversas características do aparelho auditivo humano, geram notas que simulam os resultados de testes subjetivos (ITU-T 85, 1994; ITU-T P.830, 1996a; ITU-T P.830, 1996b). O algoritmo de avaliação objetiva utilizado neste trabalho é o PESQ (Perceptual Evaluation of Speech Quality), padronizado na norma de referência P.862 da ITU-T (ITU-T P.862, 2001).

Os parâmetros de fala escolhidos para teste são parâmetros comumente utilizados em aplicações envolvendo processamento de sinais de fala, amplamente referenciados na literatura (ANDERSON, 1992; CAWLEY & NOAKES, 1993; HERNANDEZ-GOMEZ & LOPEZ-GONZALO, 1993; KITAMURA & TAKEI, 1996). São eles:

- Período esquerdo: número de amostras entre o início do quadro e a marca de pitch central.
- Período direito: número de amostras entre a marca de pitch central e o final do quadro.
- Taxa de cruzamento de zeros: contagem do número de vezes, por centésimo de segundo, em que ocorreu mudança de sinal entre amostras consecutivas do quadro.
- Taxa de máximos e mínimos: contagem do número de inflexões da forma de onda ao longo do quadro, por centésimo de segundo.
- Parâmetros mel-cepstrais: transformada cosseno (DCT) do módulo do espectro em dB do sinal. O espectro é calculado através da transformada rápida de Fourier (FFT) com 1024 pontos. Antes do cálculo do módulo do espectro em dB, os coeficientes do espectro são submetidos a uma filtragem por um banco de filtros cujo resultado é a energia distribuída em 24 bandas críticas na escala mel. A escala mel é uma transformação do eixo de freqüência, linear para freqüências baixas e aproximadamente logarítmica para freqüências altas, cujo objetivo é simular a resposta em freqüência do ouvido humano. Para o cálculo dos parâmetros mel-cepstrais considerados neste trabalho, foi utilizado o algoritmo de Davis & Mermelstein (DAVIS & MERMELSTEIN, 1980).

**Tabela 1 - Valores da avaliação objetiva para diferentes conjuntos de parâmetros.**

| Mceps 1-6 | Mceps 1-10 | Mceps 1-12 | P. Esq. | P. Dir. | Cr. Zero | Max/Min | PESQMOS |
|---|---|---|---|---|---|---|---|
| | | X | X | | | | **2358,56** |
| | | X | X | X | | | **2356,41** |
| | | X | X | | X | | **2351,67** |
| | | X | X | | X | X | **2334,32** |
| | | X | X | | | X | **2331,55** |
| | | X | | | | | **2311,15** |
| | X | | | | | | **2306,27** |
| X | | | | | | | **2165,50** |

A Tabela 1 apresenta o resultado da aplicação do algoritmo PESQ (medida PESQMOS, variando entre 0 no pior caso e 4500 no melhor) para os diferentes conjuntos de parâmetros testados. O melhor arranjo topológico obtido na análise topológica e de inicialização (descrito na subseção seguinte) foi utilizado nas

simulações. Não foi feita a normalização dos parâmetros mel-cepstrais, pois seus valores já são aproximadamente proporcionais à variância dos parâmetros. Por outro lado, os parâmetros de período foram normalizados em função da variância do primeiro parâmetro mel-cepstral. Vê-se que o melhor conjunto obtido é composto por 13 parâmetros (os 12 primeiros parâmetros mel-cepstrais mais o período esquerdo do quadro).

### 5.2. Análise topológica e de inicialização

Em busca da maximização da qualidade do sinal de fala gerado a partir do codebook, foram analisadas diversas configurações topológicas e de inicialização da rede de Kohonen. A rede foi implementada com o auxílio do toolkit SOM (Self-Organizing Map), mantido pelo grupo de Teuvo Kohonen na Helsinki University of Technology (SOM TOOLBOX). Os valores quantitativos referentes à avaliação objetiva que serão apresentados correspondem à média de 10 medidas objetivas para 10 treinamentos distintos da rede utilizando as 100 frases da base de testes.

#### 5.2.1. Número de neurônios

Os testes preliminares utilizando o método de quantização vetorial mostraram existir um compromisso entre o número de neurônios da rede e o número de clusters de saída gerado pelo algoritmo k-means, razão pela qual faz-se essencial avaliar a influência do número de neurônios para um número fixo de clusters. Para isso, treinou-se a rede com uma única frase da base, contendo 440 quadros, sendo fixado o número de clusters na saída do k-means em 440. O principal objetivo dessa análise é observar a capacidade da rede de gerar um codebook contendo representantes distintos para cada vetor de entrada, ou seja, gerar um codebook formado por todos os vetores originais. No primeiro ensaio foi utilizada uma rede de 21x21 neurônios (número de neurônios igual ao número de clusters) e no segundo ensaio utilizou-se uma rede de 50x50 neurônios (número de neurônios bem maior do que o número de clusters). O resultado da avaliação objetiva foi de 1816 para o primeiro ensaio e 3564 para o segundo. Através de inspeção auditiva, percebe-se

claramente que o sinal de fala gerado com o primeiro codebook apresenta sérias degradações, enquanto o sinal gerado com o segundo quase não apresenta degradações perceptíveis. Na seqüência dos testes, analisamos a influência do número de neurônios no treinamento com toda a base de dados (contendo cerca de 52000 quadros). A Tabela 2 mostra os resultados da avaliação objetiva obtidos para mapas com diferentes números de neurônios. É possível notar que, à medida em que se aumenta o número de neurônios, ocorre um aumento do valor do resultado da avaliação objetiva.

**Tabela 2 - Desempenho em função do número de neurônios na rede**

| Número de neurônios | PESQMOS |
|---|---|
| 25x25 - 625 | 2330,74 |
| 40x40 - 1600 | 2349,17 |
| 60x60 - 3600 | 2358,56 |

### 5.2.2. Ajuste dos neurônios

Uma característica da rede determinante para a qualidade do resultado final é a região de vizinhança em torno do neurônio vencedor dentro da qual há ajuste de pesos durante o treinamento. O objetivo dos ensaios descritos a seguir é avaliar a influência do tamanho dessa região. Três diferentes estratégias de vizinhança foram testadas. A primeira utilizou uma vizinhança denominada grande, capaz de ajustar todos os neurônios da rede. A segunda utilizou uma vizinhança pequena, com um raio abrangendo alguns poucos vizinhos. A terceira utilizou uma vizinhança decrescente, iniciando o treinamento com uma vizinhança capaz de ajustar todos os neurônios da rede e decrescendo linearmente o seu tamanho ao longo do processo, de modo a finalizá-lo com uma região abrangendo apenas os neurônios mais próximos. O resultado da avaliação objetiva foi 2241,70 para a vizinhança grande, 2429,20 para pequena e 2472,20 para decrescente. Nota-se que o erro apresentado

pela vizinhança grande é maior se comparado com os demais. A vizinhança decrescente apresentou menor erro PESQMOS.

### 5.2.3. Tamanho do codebook

Por fim, passamos à análise do tamanho do codebook de fala. O objetivo aqui é avaliar a influência do número de codevectors de saída obtidos pelo algoritmo k-means. Teoricamente, quanto maior o tamanho do codebook, maior a quantidade de vetores para representação da base inicial codificada e, portanto, maior a qualidade perceptual dos sinais reconstruídos. A comprovação exaustiva desse fato através de medidas experimentais extrapola o escopo deste trabalho.

**Tabela 3 - Desempenho do tamanho do codebook**

| Número de codevectors | PESQMOS |
|---|---|
| 20 | 2058,96 |
| 100 | 2226,89 |
| 500 | 2341,86 |
| 1000 | 2415,90 |
| 3000 | 2382,33 |

A Tabela 3 apresenta o resultado PESQMOS para distintos tamanhos de codebook. Os resultados foram obtidos utilizando a base de treinamento com 100 frases (cerca de 52000 quadros) e uma rede de 60x60 neurônios. Nota-se pela tabela que os valores PESQMOS aumentam até o codebook com 1000 codevectors, como era esperado. No entanto, para um codebook com 3000 codevectors, o valor PESQMOS é menor do que para o codebook com 1000 codevectors. Isso deve-se provavelmente ao fato de que, nesse caso, o número de neurônios na rede é

insuficiente para representar 3000 clusters. Como mostrado anteriormente, existe um compromisso entre o número de neurônios e o número de clusters.

### 5.3. Comparação com o MPEG1-Layer 3

Com o intuito de avaliar o potencial da técnica de quantização vetorial aqui proposta, comparamos os valores de avaliação objetiva para a melhor configuração encontrada na seção anterior com os valores obtidos utilizando-se o codec MPEG1-Layer 3 (MP3). Foi utilizado um codebook contendo 500 codevectors treinado a partir de 100 frases, utilizando um mapa de 60x60 neurônios, propiciando uma taxa de compressão de cerca de 100 vezes. Na compressão via MP3, utilizou-se uma taxa de bits constante (CBR) de 8 kbps, propiciando uma taxa de compressão de cerca de 30 vezes. A Tabela 4 apresenta o resultado da comparação objetiva.

Vemos que a abordagem utilizando o método de quantização vetorial apresentou uma taxa de compressão mais de três vezes superior ao do MP3 e o resultado da avaliação objetiva indicou um menor nível de degradação. Ao se fazer a inspeção auditiva dos sinais decodificados, percebe-se que as degradações impostas pelas duas técnicas são diferentes. A técnica de quantização aqui proposta introduz efeitos de descontinuidade do sinal, ao passo que a codificação MP3 tipicamente torna o sinal "abafado", indicando perda de componentes de alta freqüência.

**Tabela 4 - Comparação do método de quantização vetorial com MPEG1-Layer 3**

| Tipo de compressão | PESQMOS |
|---|---|
| Sem compressão | 4500 |
| Quantização vetorial (compr. 100 vezes) | 2343 |
| MPEG1 - Layer 3 (compr. 30 vezes) | 1904 |

## Considerações Finais

Neste trabalho, foi proposto um método de compressão de sinais de fala baseado em quantização vetorial. O método utiliza como unidades básicas do sinal de fala os quadros obtidos através da segmentação do sinal nas suas marcas de pitch. Estes quadros são parametrizados e agrupados em diferentes clusters, de acordo com as suas características acústicas.

Para o agrupamento (clusterização) dos quadros de fala, foi utilizada uma rede neural de Kohonen em conjunto com o algoritmo k-means. O uso deste último deve-se à necessidade de que o número de neurônios na rede seja superior ao número de clusters presentes no codebook de fala.

Os resultados experimentais mostraram que o método foi bem-sucedido, agrupando os quadros do sinal de fala de forma coerente. A extração dos parâmetros dos quadros é uma etapa de fundamental importância para o correto agrupamento produzido pela rede de Kohonen. Os parâmetros mel-cepstrais mostraram preservar, mais do que qualquer outro parâmetro analisado, as características acústicas da fala relevantes para o processo de clusterização. A qualidade do sinal de fala reconstruído a partir do codebook é proporcional à relação entre a quantidade de codevectors existentes no codebook e o número de quadros existentes na base de fala original.

O método de quantização vetorial aqui proposto apresentou resultados bastante promissores para aplicações de compressão de fala. Comparações com o codec MPEG1-Layer 3 (MP3) utilizando o algoritmo PESQ para avaliação objetiva da qualidade dos sinais de fala resultaram em notas maiores para o algoritmo de quantização vetorial proposto.

Nas próximas etapas deste trabalho, prevê-se a análise de outros parâmetros extraídos de sinais de fala, tais como coeficientes de predição linear (LPC), coeficientes LP-Cepstrais, coeficientes espectrais, coeficientes mel/Bark espectrais etc. Prevê-se ainda o estudo de mapas auto-construtivos, de forma a melhorar o agrupamento dos neurônios na rede de Kohonen e eliminar o uso do algoritmo k-

means. Além disso, será estudada uma forma de inicialização dos pesos da rede utilizando quadros de fala reais amostrados da base de treinamento.

## Referências bibliográficas

ANDERSON, T. R. **Phoneme recognition using an auditory model and a recurrent self-organizing neural network**. ICASSP92: IEEE International Conference on Acoustics, Speech and Signal Processing, 2:337–40, 1992.

BOLL, S. **Suppression of acoustic noise in speech using spectral subtraction**. IEEE Trans. on Acoustics, Speech, and Signal Processing, 27(2):113–120, 1979.

BUZO, A.; GRAY, A. H.; GRAY, R. M.; MARKEL, J. D. **Speech coding based upon vector quantization**. IEEE Trans. on Acoustics, Speech, and Signal Processing, 28(5):562–574, 1981.

CAWLEY, G. C.; NOAKES, P. D. **The use of vector quantization in neural speech synthesis**, volume III, Piscataway, NJ, USA. IEEE Service Center, IJCNN93, International Joint Conference on Neural Networks. 1993.

DAVIS, S.; MERMELSTEIN, P. **Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences**. IEEE Trans. on ASSP, 28(4):357–366, August 1980.

FORGEY, E. **Cluster analysis of multivariate data: efficiency vs. interpretability of classification**. Biometrics, (21):768, 1965.

GERSHO, A.; GRAY, R. **Vector quantization and signal compression**. Kluwer Academic Publishers, 2nd edition, 1992.

GRAY, R. M.; NEUHOFF, D. **Quantization**. IEEE Trans. on Inf. Theory, 44(6), 1998.

HAYKIN, S. **Neural networks, a comprehensive foundation**. Prentice Hall**,** 2nd edition, 1999.

HERNANDEZ-GOMEZ, L. A.; LOPEZ-GONZALO, E. **Phonetically-driven CELP coding using self-organizing maps**, volume II, Piscataway, NJ. IEEE Service Center. ICASSP93, International Conference on Acoustics, Speech Propagation and Signal Processing. 1993.

ITU-T. 85 – **A method for subjective performance assessment of the quality of speech voice output devices**, June 1994.

ITU-T. P.830 – **Subjective performance assessment of telephone-band and wideband digital codecs**, February 1996a.

ITU-T. P.830 – **Subjective performance assessment of telephone-band and wideband digital codecs**, February 1996b.

ITU-T. P.861 – **Objective quality measurement of telephone-band (300-3400 Hz) speech codecs**, February 1998.

ITU-T. P.862 – **Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrow band telephone networks and speech codecs**, February 2001.

KITAMURA, T.; TAKEI, S. **Speaker recognition model using two-dimensional mel-cepstrum and predictive neural network**, volume 3, New York, NY, USA. Proceedings ICSLP 96. Fourth International Conference on Spoken Language Processing. 1996.

KOHONEN, T. **Self-organized formation of topologically correct feature maps**. Biological Cybernetics, (43):59–69, 1982.

KOHONEN, T. **Improved versions of learning vector quantization**. IJCNN International Joint Conference on Neural Networks, (1):545–550, 1990a.

KOHONEN, T. **The self-organizing map**. Proceedings of the IEEE, (78):1464–1480, 1990b.

KRISHNAMURTHY, A.; AHALT, S.; MELTON, D.; CHEN, P. **Neural networks for vector quantization of speech and images**. IEEE Journal on Selected Areas in Communications, 8(8):1449–1457, 1990.

LLOYD, S. P. **Least squares quantization in PCM**. IEEE Trans. Information Theory, (28):129–137, 1982.

MACQUEEN, J. **Some methods for classification and analysis of multivariate observations**. Proc. Fifth Berkeley Symp. Math. Statistics and Probability, (1):281–296, 1967.

MAKHOUL, J.; WOLF, J. **Linear prediction and the spectral analysis of speech**. Bolt, Beranek, and Newman Inc., pages 172–185, 1972.

RUNSTEIN, F. **Sistema de reconhecimento de fala baseado em redes neurais artificiais**. Tese de Doutorado, FEEC/Unicamp, 1998

SOM TOOLBOX. Disponível em: http://www.cis.hut.fi/projects/somtoolbox/.

ULTSCH, A.; SIEMON, H. P. **Kohonen's self-organizing feature maps for exploratory data analysis**. In Proc. INNC'90, Int. Neural Network Conf., 305-308 edition, 1990.