



Faculdades Adamantinenses Integradas (FAI)

[www.fai.com.br](http://www.fai.com.br)

LIMA, Fernando Parra dos Anjos; SILVA, James Clauton; ESTEVAM, Giuliano Pierre; MINUSSI, Carlos Roberto. Redes neurais artificiais aplicadas no reconhecimento de tons DTMF. Omnia Exatas, v.3, n.1, p.79-89, 2010.

## REDES NEURAIS ARTIFICIAIS APLICADAS NO RECONHECIMENTO DE TONS DTMF

### ARTIFICIAL NEURAL NETWORKS APPLIED IN THE RECOGNITION OF DTMF TONES

**Fernando Parra dos Anjos Lima**

**James Clauton da Silva**

Graduando de Engenharia da Computação – Unisalesiano - Araçatuba – SP

**Giuliano Pierre Estevam**

Professor Titular de Engenharia – Unisalesiano - Araçatuba – SP

**Carlos Roberto Minussi**

Prof. Titular do Dep. Engenharia Elétrica - UNESP – Ilha Solteira – SP

## RESUMO

Este plano de pesquisa visa um estudo dos principais conceitos e arquiteturas das Redes Neurais Artificiais, e um desenvolvimento prático com a aplicação destes conceitos. Para o desenvolvimento prático presente neste estudo utiliza-se um problema proposto em referência bibliográfica, assim implementa-se uma aplicação na plataforma de desenvolvimento Borland Delphi 7 para reconhecimento e classificação de tons DTMF (Dual Tone Multi Frequency) proposta pelo autor Luciano Frontino de Medeiros, no livro "Redes Neurais em Delphi". Cada dígito correspondente a um tom DTMF é gravado então em um arquivo que será utilizado para o treinamento da rede SOM. No entanto, o sinal puro do tom DTMF não pode treinar uma rede SOM. Primeiro, o sinal deve ser analisado por um algoritmo FFT, que permite identificar as frequências que compõem o tom DTMF. A partir dessas frequências, o conjunto de componentes é usado para compor o vetor de entrada da rede SOM. A FFT é calculada e obtém-se o espectro das frequências do sinal. Cada espectro obtido é usado para o vetor de entrada da rede SOM. Portanto, para o treinamento da rede SOM, cada dígito será processado a partir de um arquivo. Após o treinamento, o teste da rede será feito usando-se tons DTMF compostos tal como em uma discagem qualquer. Para evidenciar o potencial de aprendizado da rede SOM para esse exemplo, na fase de treinamento serão utilizados tons DTMF gerados no próprio aplicativo. Na fase de teste, os tons DTMF terão origem diferente. O reconhecimento de tons DTMF será feito a partir de um algoritmo que considera o treinamento da rede SOM de forma individualizada, considerando cada tom como um *cluster*. Assim podendo realizar a identificação do tom através da amostragem FFT gerada e comparar com o conhecimento obtido em fase de treinamento. Para cada amostra apresentada na fase de teste é gerado um valor numérico com a identificação do tom DTMF da respectiva amostra.

**Palavras-chaves:** Redes Neurais Artificiais, Reconhecimento de Padrões, Tons DTMF.

## ABSTRACT

This research plan aims to study the main concepts and architectures of neural networks, and developing a practice in applying those concepts. To develop this practice is used in this study a problem proposed in reference literature, so we implement an application development platform in Borland Delphi 7 for recognition and classification of DTMF (Dual Tone Multi Frequency) proposed by the author Luciano de Medeiros Frontino in the book "Neural Networks in Delphi." Each digit corresponds to a DTMF tone is then recorded in a file that will be used for training the SOM. However, the pure tone DTMF signal can not train a SOM. First, the signal must be analyzed by an FFT algorithm that can identify the frequencies that make up the DTMF tone. From these frequencies, the number of components is used to compose the input vector of SOM. The FFT is calculated and we obtain the spectrum of signal frequencies. Each spectrum obtained is used for the input vector of SOM. So for training the SOM, each digit is processed from a file. After training, the

network testing will be done using DTMF tones compounds as in any one dial. To enhance the learning potential of the SOM for this example, the training phase will be used DTMF tones generated by the application itself. In the test phase, the DTMF tones will have different origin. The recognition of DTMF tones will be based on an algorithm that considers the SOM training individually, considering each tone as a cluster. So can make the identification of tone FFT generated by sampling and comparing them with the knowledge gained in training. For each sample presented in the test phase is generated with a numerical value identifying the DTMF tone of their sample.

**Key-words:** Artificial Neural Networks, Pattern Recognition, DTMF tones.

## INTRODUÇÃO

Redes Neurais Artificiais são conceitos do mecanismo de funcionamento do cérebro aplicados em software e/ou hardware, por tanto são capazes de adquirir conhecimento, ou seja, aprenderem com a experiência. As redes são capazes de efetuar diagnosticos complexos, sendo assim as redes se modelam através de varias unidades de neurônios ou unidades de processamento, dispostos em fileiras ou camadas, formando um arranjo de interligações, tais interligações são conexões constituídas por pesos, ou sinapses, que são o ponto de calibração das unidades de processamento. O processo de calibração dos pesos é chamado de treinamento ou aprendizado, o mesmo deve ser elaborado no modo off-line do sistema, ou como podemos dizer, antes que o sistema entre em funcionamento. Uma vez ajustada, empregando-se algum tipo de algoritmo de adaptação dos referidos pesos, a rede se denomina apta para efetuar diagnosticos e emitir um grau de precisão satisfatorio no reconhecimento de padrões não utilizados no seu processo de aprendizado.

Este processo pode ser visto como uma possibilidade na viabilização de rotinas de análises em tempo real, que certamente é importante no diagnostico de varias areas do conhecimento humano, tais como: medicina, biologia, engenharia, ecônomia entre outras.

Levando em consideração a importância das análises de rotinas em tempo real, este plano de pesquisa visa realizar um estudo embasado nos conceitos e arquiteturas neurais artificiais e suas aplicações de modo geral, e em específico o reconhecimento de padrões. Utilizaremos a plataforma e linguagem de programação Borland Delphi 7 para implementação de um sistema para reconhecimento de som com redes neurais artificiais. Na linguagem de programação Borland Delphi 7 será utilizado alguns componentes que simulam a arquitetura e toda a técnica da rede SOM (Self Organizing Maps), ou mais conhecidas como mapas auto-organizáveis proposto por Kohonen.

Na rede de reconhecimento de som utilizaremos o conceito de tons DTMF (Dual Tone Multi Frenquency) para gerar amostras de som para utilização no processo de treinamento e reconhecimento, ou seja, em questão temos tons sonoros reproduzidos apartir de um teclado de telefone.

## MATERIAIS E MÉTODOS

Para implementar o sistema utiliza-se a plataforma de desenvolvimento Borland Delphi 7, e alguns componentes prontos que ajudaram nas tarefas a serem desenvolvidas. Utilizaremos os seguintes componentes:

- ✓ TSOM: O objeto TSOM é uma implementação do conceito de mapas auto-organizáveis (do inglês Self Organizing Maps, ou de forma simplificada, SOM, ou mesmo redes SOM), o modelo proposto por Kohonen, também chamado de redes de Kohonen, esse modelo estabelece um mapeamento topológico uni ou bidimensional a partir de um conjunto de dados de dimensão arbitraria, e a formação do mapa é realizada de forma adaptativa.
- ✓ TFFT: O objeto TFFT é a implementação do conceito de processamento de sinais digitais que faz a transformada rápida de Fourier, podendo fazer a componente direta ou inversa.

- ✓ TWaveHandle: O objeto TwaveHandle é a implementação que permite trabalhar manipular e criar arquivos no formato WAVE.

Os componentes apresentados anteriormente fazem o processamento de sinais digitais, incluindo a transformada direta e indireta de Fourier, permitem o trabalho, manipulação e criação de arquivos com o formato WAVE, e principalmente a implementação do conceito de mapas auto-organizáveis, ou ainda mais conhecidos como rede de Kohonen.

Como o sistema trabalha com arquivos do tipo WAVE, a seguir vamos demonstrar como é o funcionamento do arquivo WAVE.

### Manipulação de arquivos WAVE, Formato RIFF

Passamos a explicação do componente TwaveHandle, que é responsável pela manipulação de arquivos WAVE. Em alguns momentos será preciso carregar ou salvar arquivos nesse formato para que possamos treinar a rede SOM. Tais arquivos seguem o padrão RIFF (Resource Intechange File Format), da Microsoft. Esse padrão é aplicável também a outros tipos de formatos de arquivos, como o padrão MIDI, e a sua estrutura será dividida em pedaços ou chunks. Os chunks utilizados no componente TwaverHundle são:

- ✓ RIFF Header Chunk.

Refere-se ao cabeçalho que identifica o arquivo RIFF. A Tabela 1 apresenta sua estrutura.

Tabela 1: estrutura do cabeçalho de identificação do arquivo RIFF.

Descrição	Tamanho (bytes)	Conteúdo
Identificação	4	'riff'
Comprimento do arquivo (inclusive o cabeçalho)	4	
Identificação do tipo de mídia	4	'Wave'
<b>Total</b>	<b>12</b>	

- ✓ Format Chunk.

Retém as informações de formato do arquivo. Possui a estrutura apresentada na Tabela 2.

Tabela 2: Estrutura do formato no arquivo RIFF.

Descrição	Tamanho (bytes)	Conteúdo
a) Identificação do formato	4	'fmt'
b) Comprimento da descrição	4	
c) Código de compressão	2	
d) Número de canais	2	1 ou 2
e) Taxa de amostragem (bytes por segundo)	4	
f) Média de bytes por segundo	4	
g) Alinhamento de bloco	2	(d . h) / 8
h) Bits por amostra	2	8 ou 16
<b>Total</b>	<b>24</b>	

O comprimento da descrição deverá conter o tamanho total do cabeçalho (16 bytes) mais alguma informação extra. O número de canais pode assumir os valores 1 (para um canal mono) ou 2 (para um canal estéreo). A taxa de amostragem é medida em hertz (Hz) e indica a qualidade de amostragem dos dados. Os valores padrão são 8000, 11025, 22050 e 44100. A quantidade de bits por amostra pode assumir 8 ou 16 bits. O alinhamento de bloco é calculado tomando-se o número de canais multiplicado pela quantidade de bits por amostra e dividido por 8.

✓ Data Chunk.

Refere-se aos blocos de dados. Possui a estrutura apresentada na Tabela 3.

Tabela 3: estrutura dos blocos no arquivo RIFF.

Descrição	Tamanho (bytes)	Conteúdo
Identificação	4	'data'
Comprimento dos dados	4	

✓ Fact Chunk

Esse chunk é de uso opcional, porém foi colocado no componente para manter compatibilidade com arquivos WAVE já existentes que podem contê-lo. Nesse chunk é gravado o código de compressão dependendo do conteúdo do arquivo WAVE. Como para a nossa finalidade estaremos trabalhando com dados sem compressão, seu uso será suprimido (apenas mantido na leitura do arquivo).

✓ Dados

Um arquivo WAVE possui a estrutura genérica de chunks, demonstrada na tabela 4.

Tabela 4: Estrutura genérica de um arquivo WAVE.

Reader RIFF
Format Chunk
Data Chunk
Dados
...
Dados

Após o Data Chunk, temos os dados propriamente ditos, que são gravados em formato PCM (Pulse Code Modulation). Os bytes de dados serão armazenados de acordo com o número de bits por amostra e o número de canais. O tipo *TarrayWave* é definido com um vetor de tamanho 32768 de tipo `smallint`, que pode gravar 2 bytes para um valor. O arquivo WAVE será lido e os dados armazenados em buffers de tipo *TarrayWave*.

O número de bits por amostra indica a faixa de valores possíveis que um certo sinal pode assumir. Desde que estejamos trabalhando com um dado do tipo `smallint`, se o número de bits por amostra for 8, os valores irão de -128 a 127. Porém, se for de 16 bits, a faixa poderá assumir valores de -32768 a 32767, permitindo uma melhor quantização do sinal. Em nosso caso, estaremos trabalhando com sinais de um canal apenas e com 16 bits por amostra. Assim, cada conjunto de 2 bytes teremos um dado sinal no buffer de dados.

Após compreender o funcionamento do arquivo WAVE, e conhecer o conceito de cada componente que será utilizado no sistema, passamos a frente, e vamos à descrição do aplicativo.

### Modelagem do aplicativo

Antes de mostrar as telas do software, vamos definir a rede neural que será utilizada. Para o sistema utiliza-se uma rede neural de Kohonen, mais conhecida como rede SOM, uma rede com duas camadas. Uma camada de entrada e outra camada de saída.

Cada dígito correspondente a um tom DTMF é gravado então em um arquivo que será utilizado para o treinamento da rede SOM. No entanto, o sinal puro do tom DTMF não pode treinar uma rede SOM. Primeiro, o sinal deve ser analisado por um algoritmo FFT, que permite identificar as freqüências que compõem o tom DTMF. A partir dessas freqüências, o conjunto de componentes é usado para compor o vetor de entrada da rede SOM.

Um tom DTMF gravado em um arquivo terá certa duração. O calculo da FFT de sinal deve considerar um vetor de largura constante, e esse vetor é preenchido como se uma “janela” fosse se deslocando sobre o sinal. A cada janela, a FFT é calculada e obtém-se o espectro das freqüências do sinal. Cada espectro obtido é usado para o vetor de entrada da rede SOM.

Portanto, para o treinamento da rede SOM, cada dígito será processado a partir de um arquivo. Após o treinamento, o teste da rede será feito usando-se tons DTMF compostos tal como em uma discagem qualquer. Para evidenciar o potencial de aprendizado da rede SOM para esse exemplo, na fase de treinamento serão utilizados tons DTMF gerados no próprio aplicativo. Na fase de teste, os tons DTMF terão origem diferente.

O reconhecimento de tons DTMF será feito a partir de um algoritmo que considera o treinamento da rede SOM de forma individualizada, considerando cada tom como um *cluster*. Após esse treinamento, os *clusters* irão compor uma única rede SOM.

### Telas do sistema

Inicialmente temos a tela principal, com os menus de acesso as funções. A Figura 1 ilustra este processo.

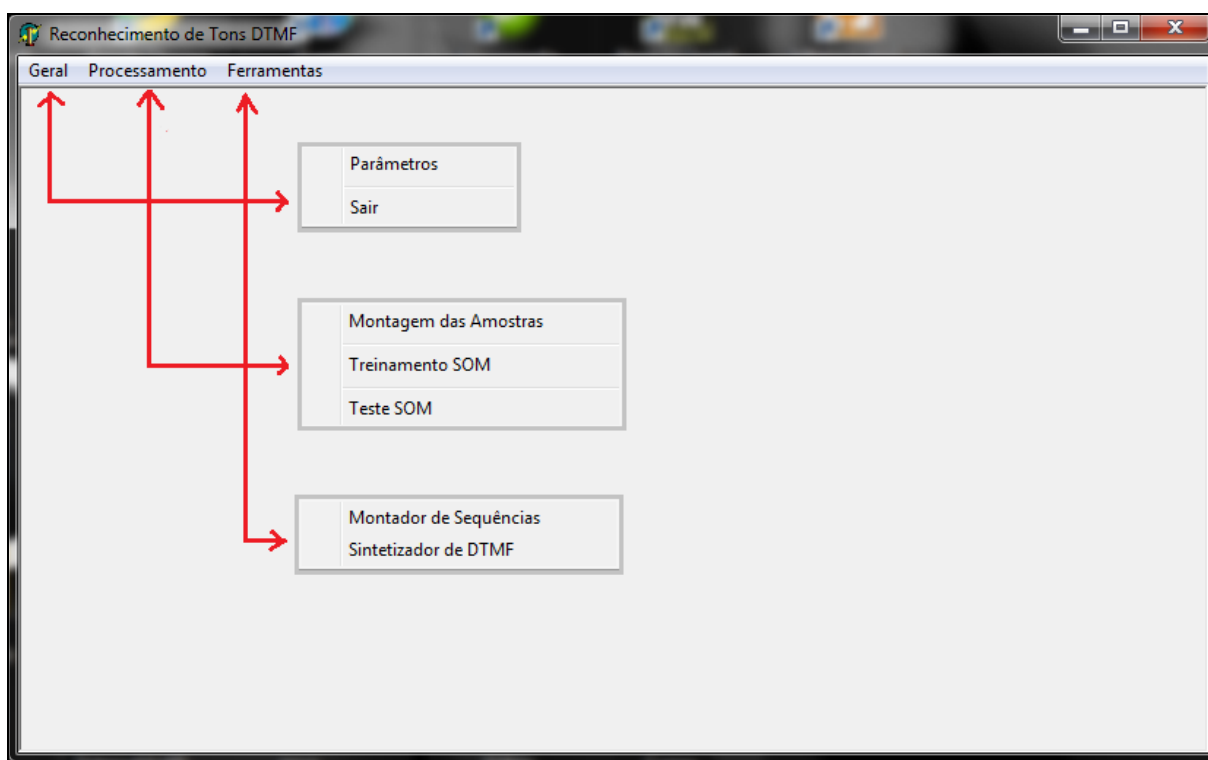


Figura 1: Tela principal do sistema e os menus.

Para o nosso propósito, um tom DTMF deverá ser capturado e gravado em arquivo de forma que se possa analisá-lo posteriormente. Para isso no sistema existem duas ferramentas para sintetizar e gerar tons DTMF para se trabalhar no aplicativo.

A Figura 2 mostra a tela do gerador de tons DTMF:

Este tom gerado pelo sintetizador é armazenado em um arquivo do tipo WAVE, e pode ser utilizado para fazer treinamento e teste no sistema.

Alem de podermos trabalhar com um único tom também podemos trabalhar com seqüências de tons, para isso temos a segunda ferramenta, um sintetizador de seqüências de tons DTMF, a Figura 3 mostra a tela do sintetizador de seqüência:

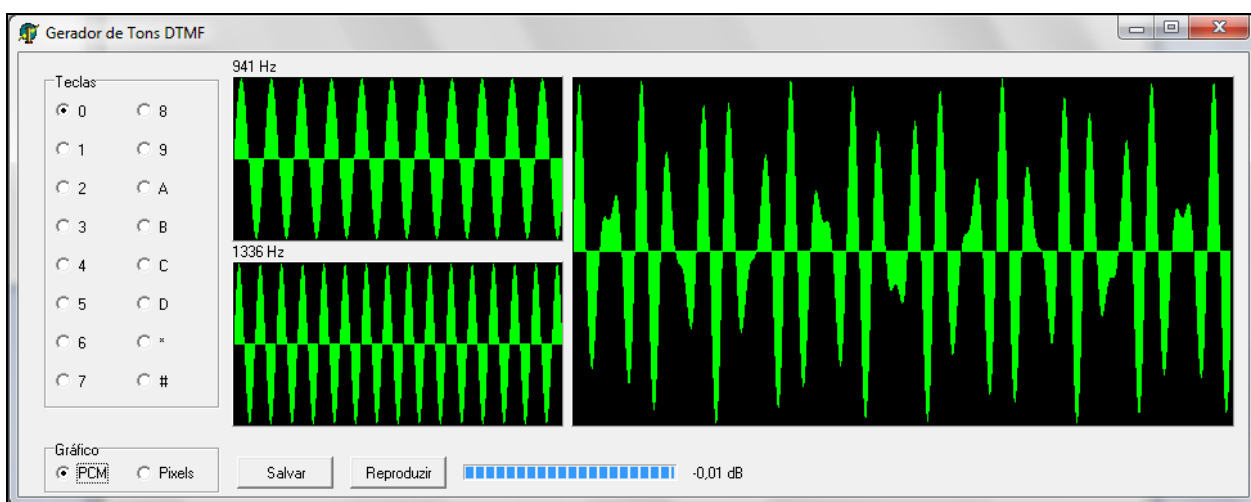


Figura 2: Gerador de tons DTMF, tom zero sendo gerado.

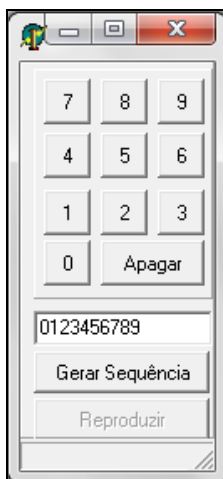


Figura 3: Sintetizador de seqüências, gerando a seqüência "0123456789".

Finalmente após realizarmos o processo de geração de tons DTMF, podemos mostrar o processo de funcionamento da rede.

Primeiramente precisamos definir os parâmetros que a nossa rede vai ter. A Figura 4 demonstra este processo:

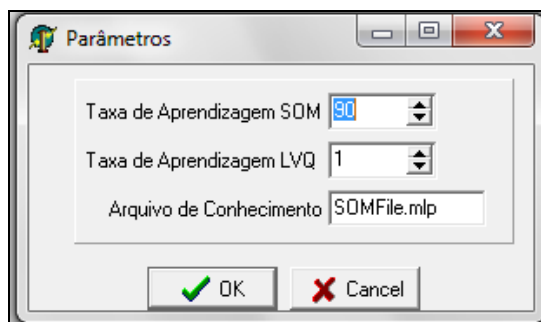


Figura 4: Tela de parâmetros.

Na tela de parâmetros nós definimos qual a taxa de aprendizagem da rede SOM, que varia de 0 a 100%, definimos a taxa de aprendizagem do algoritmo de treinamento LVQ (Learning Vector Quantization), e o nome do arquivo de conhecimento da rede.

Depois de realizado a configuração dos parâmetros, passamos ao processamento, no menu processamento encontramos a opção montagem das amostras. Na etapa de montagem das amostras será realizado um processamento com o arquivo WAV para gerar sua amostragem em uma componente FFT, atribuída pela transformada rápida de Fourier. A Figura 5 ilustra a tela do software onde temos a representação da amostra e sua componente FFT sendo gerada:

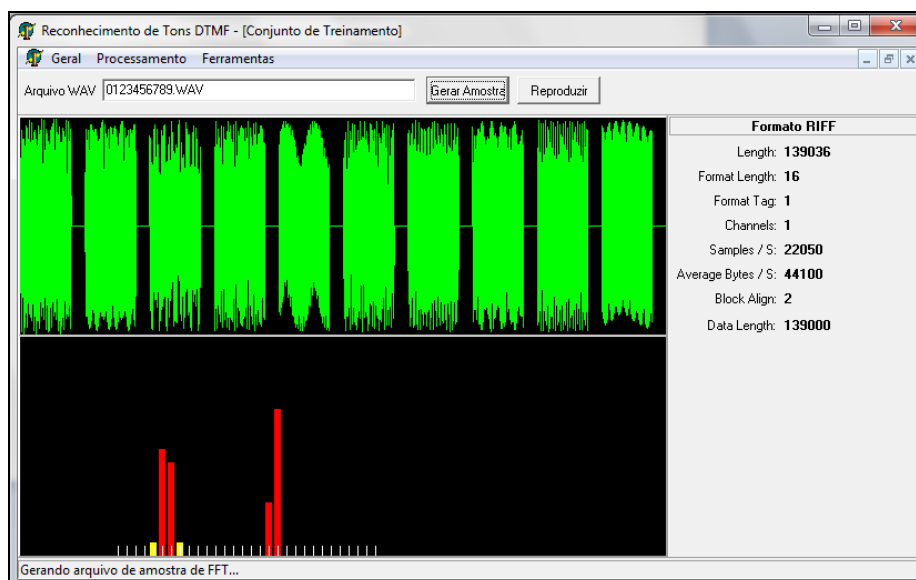


Figura 5: Gerando amostra para entrada na rede SOM.

No processo de montagem de amostra é gerado o componente FFT que será o vetor de entrada da rede SOM.

Após realizar o processo de amostragem, podemos treinar a rede para obter o conhecimento sobre os respectivos padrões de amostragem apresentados na entrada.

A tela apresentada na Figura 6 mostra o algoritmo LVQ em processamento:

No processo de treinamento a rede obtém conhecimento de cada espectro gerado pela componente FFT e armazena estes espectros em arquivo de texto, na imagem podemos observar que para cada dígito do teclado temos uma faixa de espectro. Assim a rede pode guardar um padrão sobre os mesmos em seu arquivo de conhecimento para futuramente identificar um espectro parecido ou igual.

Após realizar o treinamento temos que testar a rede para obtermos os resultados e analisar a sua eficiência.





Figura 6: treinamento sendo executado.

A Figura 7 a seguir ilustra o processo de reconhecimento e classificação dos tons DTMF. Temos uma tela que ilustra a amostra em forma de gráfico, temos as propriedades do arquivo WAVE, a amostragem da FFT, e temos o local onde será exibido o valor do dígito do tom DTMF.

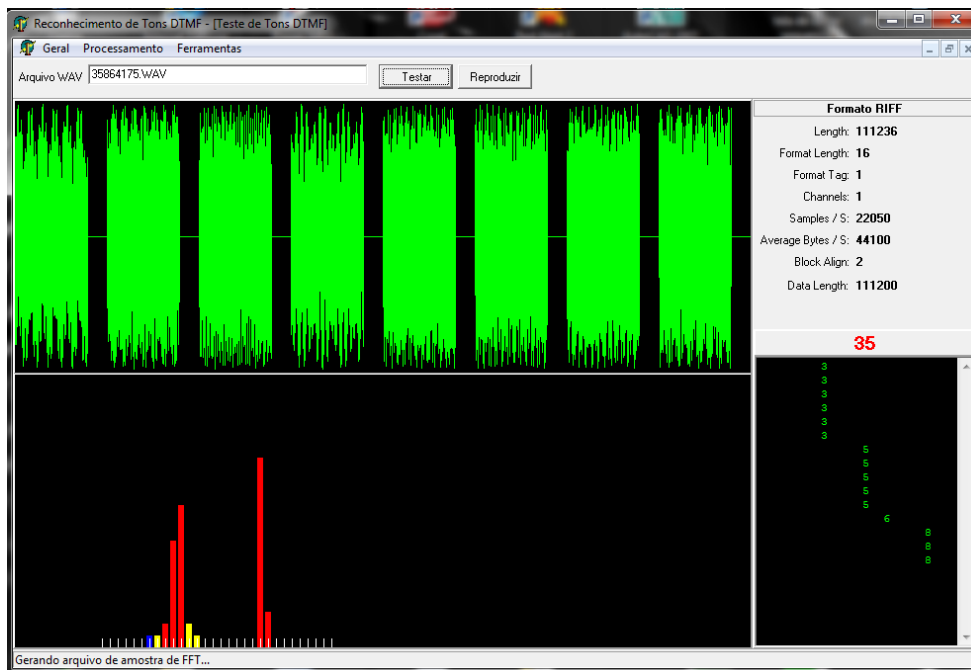


Figura 7: Teste da rede em execução, reconhecimento de uma amostra em seqüência.

O funcionamento do teste tem o mesmo conceito do treinamento, primeiramente fazemos a amostragem FFT do arquivo WAVE e depois comparamos esta amostragem com os padrões armazenados no arquivo de conhecimento da rede SOM, assim a rede pode identificar os tons DTMF, imediatamente a cada amostragem FFT gerada é feito o processo de identificação pela rede e exibido qual é em questão o valor do Tom DTMF amostrado.

Então no final do treinamento temos uma seqüência numérica que corresponde ao valor da amostra, ou seja, temos a identificação dos tons e sabemos qual número representa cada som amostrado na seqüência.

## RESULTADOS E DISCUSSÕES

Para o aplicativo fizemos diversos testes com amostras diferentes, utilizamos amostras sintetizadas no próprio sistema para fazer o processo de treinamento e amostragem, e também utilizamos amostras que não foram geradas no sintetizador para fazer o teste.

Em um teste realizado utilizamos uma amostra gerada no sintetizador com a seqüência "0123456789.wav" para realizar o treinamento, e depois fizemos testes com 10 amostras diferentes que não foram geradas pelo aplicativo, as amostras foram às seguintes: "11586447.wav", "15649782.wav", "20436587.wav", "20439585.wav", "22475001.wav", "35698741.wav", "35864175.wav", "42285660.wav", "48596217.wav" e "50213274.wav".

Os resultados obtidos no teste destes padrões eram esperados, pois temos arquivos de ótima qualidade, possibilitando uma boa amostragem. Dos 10 padrões apresentados para a rede não obtivemos nenhum erro, isso já era esperado. Conseqüentemente uma rede que auto se organiza tem o potencial para não ter erro, assim esperávamos o resultado positivo na identificação de padrões desconhecidos.

## CONCLUSÃO

Obtivemos resultados esperados em nossa análise e concluímos que uma rede que trabalha com o conceito de auto-organização tem um potencial enorme para trabalhar com reconhecimento de SOM.

A utilização do conceito de redes neurais artificiais no processo de identificação de sons é interessante, pois temos uma precisão maior, e também temos uma adaptação para correção de erros, a partir de treinamentos mais longos e precisos realizados pela rede. Podemos dizer que o uso da rede neural da uma segurança maior ao processo de reconhecimento porque temos um procedimento de identificação adaptativo.

Uma observação a ser feita em relação ao sistema é que quanto melhor for o arquivo WAVE amostrado para a rede no processo de treinamento, melhor será o resultado obtido no processo de teste, pois vamos obter uma amostragem FFT com padrões excelentes e quando formos realizar algum teste, mesmo que a amostra a ser testada seja de menor qualidade, vamos obter um desempenho favorável da rede neural artificial.

## REFERÊNCIAS BIBLIOGRÁFICAS

BRAGA, A. DE P.; CARVALHO, A. P. L.; LUDERMIR, T. B.. **"Redes Neurais Artificiais - Teorias e Aplicações"**, LTC, 2<sup>nd</sup> Edição, Rio de Janeiro, Brasil, 2007.

CARPENTER, G. A.; GROSSBERG, S.; MARKUZON, N.; REYNOLDS, J. H.; ROSEN, D. B. "Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps", **IEEE Transactions on Neural Networks**, September-92, Vol. 3, Number 5, pp. 698-713.

HAYKIN, S. **"Neural Networks: A Comprehensive Foundation"**, Prentice-Hall, Upper Saddle River, New Jersey, USA, 1994.

HOPFIELD, J.J. "Neural Networks And Physical Systems With Emergent Collective Computational Abilities", **Proceedings of the National Academy of Sciences of the U.S.A.**, vol. 79, pp. 2554-2558, 1982.

JAIN, A. K.; MAO, J.; MOHIUDDIN, K. M. "Artificial Neural Networks: A Tutorial", **IEEE Computer**, March 1996, Vol. 29, No. 3, pp. 31-44.

KARTALOPOULOS, S. V. **“Understanding Neural Networks And Fuzzy Logic: Basic Concepts and Applications”**, IEEE Press, Piscataway, NJ, USA, 1996.

KOHONEN, T. **“Self-Organization And Associative Memory”**, Springer-Verlag, 2<sup>nd</sup> Edition, Berlin, Germany, 1988.

KOHONEN, T. **“The Self-Organizing Map”**, **Proceedings of IEEE**, September 1990, pp. 1464-1480.

MEDEIROS, L. F. DE. **“Redes Neurais em Delphi”**, Visual Books, 2<sup>nd</sup> Edição, Florianópolis, Brasil, 2006.

NASCIMENTO, J.; CAIRO L. **“Inteligência Artificial em controle e automação”**, Edgard Blucher, 1<sup>nd</sup> Edição, São Paulo, Brasil, 2004.

WERBOS, P. J. **“Beyond Regression: New Tools For Prediction And Analysis in The Behavioral Sciences”**, PhD. Thesis, Harvard University, 1974.

WIDROW, B.; LEHR, M.A. **“30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation”**, **Proceedings of the IEEE**, pp. 1415-1442, Vol. 78, n<sup>o</sup> 9, Sep. 1990.