

OTIMIZANDO REDES NEURAS ARTIFICIAIS COM TRANSFORMADAS WAVELETS

HELTON M. PEIXOTO, JOILSON B. ALMEIDA REGO, ANA M. G. GUERREIRO, ADRIÃO D. D. NETO

Universidade Federal do Rio G. Norte Caixa Postal 1524 - Campus Universitário Lagoa Nova - UFRN
DCA – Departamento de Engenharia de Computação e Automação

E-mails: helton.maia@gmail.com, jotarego@gmail.com, anamaria@dca.ufrn,
adriao@dca.ufrn.br

Resumo—O objetivo deste artigo é apresentar o uso de transformadas wavelets na compressão e extração de características presentes em imagens, possibilitando com isso otimizar o reconhecimento de padrões utilizando redes neurais artificiais. A partir da análise em multi-resolução dos coeficientes gerados pela wavelets, é possível gerar um vetor descritor da imagem, contendo suas características mais relevantes. Dessa forma, utilizando tais características para aprendizagem de uma rede neural artificial de multicamadas, treinada utilizando o algoritmo *Resilient Backpropagation*, foi possível diminuir de forma muito eficiente o custo computacional para o funcionamento do sistema, mantendo a generalização acima de 90% para os padrões de imagens utilizados.

Palavras-chave—Wavelets, rede neural artificial, *resilient backpropagation*.

Abstract— The objective of this paper is to use wavelet transforms for the compression process and extraction of features presented in images, in order to optimize the image pattern recognition through artificial neural networks. From the multiresolution analysis of the coefficients generated by the wavelets, it is possible to generate an array descriptor of the image, with its most important features. The use of such features in the learning process of a multilayer artificial neural network, trained with the resilient backpropagation algorithm, enabled to minimize efficiently the computational cost of the process, when more than 90% of generalization was observed for all the patterns used.

Keywords— Wavelets, artificial neural network, resilient backpropagation.

1 Introdução

As *wavelets* e as redes neurais artificiais (RNA's) têm gerado enorme interesse nos últimos anos, tanto em áreas científicas como em aplicações práticas. A grande vantagem do uso das *wavelets* encontra-se no fato destas funções apresentarem um comportamento local, não só no domínio da frequência como também no domínio espaço/tempo. As RNA's possuem a capacidade de aprendizagem a partir de um conjunto de treinamento adequado, tornando-a bastante útil em muitas aplicações, principalmente no reconhecimento de padrões.

A proposta deste trabalho é a utilização da transformada *wavelet* na compressão e extração de características presentes nas imagens, dessa forma, cria-se um vetor de descritores que deve ser utilizado para otimizar o reconhecimento de padrões por uma rede neural.

O vetor de descritores contém elementos cujos valores descrevem apuradamente o conteúdo da imagem, e deve ocupar menos espaço que a imagem representada *pixel* à *pixel*. A maior dificuldade encontrada neste processo é a geração deste vetor, onde o conteúdo de interesse da imagem deve ser bem descrito, de tal sorte que se consiga obter características relevantes das imagens.

Desta forma, neste trabalho, é mostrado o uso de uma técnica de multi-resolução, a transformada *wavelet*, que efetua a filtragem, compressão e extração dos descritores de um pequeno conjunto de imagens para posterior classificação através de uma RNA.

O trabalho está organizado da seguinte forma: na seção 02, será exposta de forma resumida a teoria de *wavelets*; na seção 03, será descrita a RNA; na seção 04, descreve-se o sistema; na seção 05, apresentam-

se as simulações e os resultados obtidos, seguidos da conclusão e das referências.

2 Wavelets

2.1 Análise Multi-resolução

A análise multi-resolução é uma estratégia de processamento de sinais onde é utilizado um conjunto de filtros especializados em extrair as informações de sinal, como as frequências nele presentes e a localização das mesmas em função do tempo de duração do sinal, em diferentes resoluções (Castleman, 1996).

A breve descrição da análise multi-resolução permite apresentar as duas funções responsáveis pela geração de todo o sistema de *wavelets*: a *função escala* e a *wavelet primária* (ou *wavelet mãe*). O termo mãe vem do fato de que funções com diferentes tamanhos são usadas no processo da transformada e todas são originadas de uma *wavelet* principal, a *wavelet mãe*.

As funções de escala $\phi_{j,k}$ e as *wavelets*, $\psi_{j,k}$ dizem-se ortogonais por respeitarem a seguinte condição:

$$\int_{-\infty}^{+\infty} \phi_{j,k}(x) \psi_{j,k}(x) dx = 0 \quad (1)$$

em que, $j \in Z$ corresponde ao parâmetro que representa a *escala* a que a função está representada, e $k \in Z$ corresponde à translação de $k/2^j$ em relação à função de escala e a *wavelet* primária, dados por $j = 0$ e $k = 0$. Tanto a função de escala como a *wavelet* são definidas no conjunto dos reais (R), através de *escalamento* e *translações* das funções apresentadas.

O parâmetro *translação* corresponde à informação de tempo no domínio da transformada e o parâmetro *escalonamento* é o processo de compressão e dilatação do sinal (Mallat, 1999). Na Fig. 1, tem-se um exemplo de uma função de escala e uma *wavelet* primária de Haar.

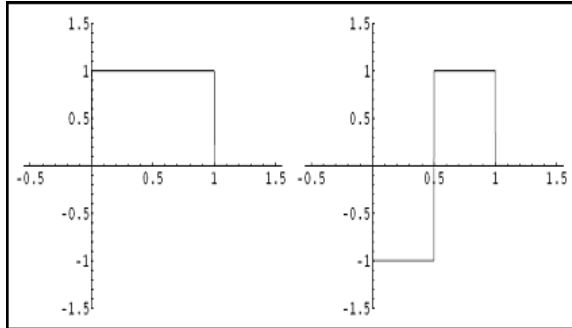


Fig. 1. Função de escala e *wavelet* primária de Haar.

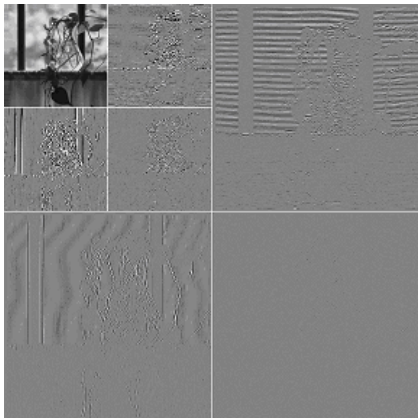


Fig. 2. Representação multi-resolução de uma imagem.

Analisando a Fig. 2, percebe-se que esta é composta por uma imagem inicial, extraída de (Gonzalez, 2004), e do grau de refinamento aplicado na mesma, através da transformada *wavelet*. A informação necessária para se obter a imagem inicial do vaso a partir da mesma, a uma *escala* menor, encontra-se nos 06 (seis) quadrados adjacentes à imagem. A informação contida nestes quadrados designa-se *detalhe ou resolução*, que é a informação necessária para passar de um grau de refinamento, ou grau de “nitidez”, para outro. Ao somar as informações referentes aos quadrados pode-se assim recompôr a imagem. Esta forma de decompor e recompôr imagens pode ser implementada com rapidez e eficácia, graças ao uso das transformadas *wavelets*.

2.2 Wavelets

As transformadas matemáticas são utilizadas em um conjunto de dados para se obter informações adicionais não disponíveis no modelo de dados primitivos. Neste caso, é necessário o uso de uma transformada que detecte a variação dos tons de cores da vizinhança de um *pixel* e a localização espacial deste, e transponha isto de forma eficiente em um espaço multi-resolução (Castleman, 1996).

Métodos tradicionais de análise de sinal, baseados na transformada de Fourier, podem determinar todas as frequências presentes no sinal, porém sua relação com o domínio temporal inexistente. Para superar este problema, surgiu a transformada de Gabor (ou *STFT – Short Time Fourier Transform*); a idéia principal desta transformada é introduzir um novo parâmetro de frequência local como se a “transformada local” observasse o sinal através de uma curta janela dentro da qual o sinal permanece aproximadamente estacionário (Oliveira, 2007). Os problemas na resolução do tempo e da frequência são resultados de um fenômeno físico conhecido como princípio da incerteza de Heisenberg (em um sinal, é impossível conhecer a frequência e o tempo exatos de ocorrência desta). Este fenômeno é indiferente em relação a transformada utilizada (Oliveira, 2007).

A transformada *Wavelet* foi desenvolvida como uma alternativa à transformada de Gabor para solucionar o problema da resolução. As *Wavelets* são funções matemáticas que separam sinais em diferentes componentes e extraem cada componente com uma resolução adequada à sua escala. Possuem vantagem em relação à transformada de Fourier, porque analisam o sinal em escalas diferentes e se deslocam analisando cada ponto do sinal. A Transformada *Wavelet* Contínua (CWT) pode ser expressa na forma:

$$CWT(\tau, a) = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{a}} \psi^* \left(\frac{t-\tau}{a} \right) dt \quad (2)$$

em que τ e a são os parâmetros translação e escala, respectivamente.

2.2.1 Wavelet Discreta

A transformada de *wavelet* contínua é calculada fazendo translações e escalonamentos contínuos de uma função sobre um sinal. Na prática, esta transformação não é viável, pois requer a realização de infinitas translações e escalonamentos, demandando muito tempo, esforço computacional e redundância. As *wavelets* discretas foram introduzidas para superar este obstáculo e por isso serão utilizadas na realização do presente trabalho. As mesmas, não são transladadas nem escalonadas continuamente, mas sim em intervalos discretos, o que é conseguido a partir de uma modificação na *wavelet* contínua:

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{|s|}} \psi \left(\frac{t-\tau}{s} \right) \quad (3)$$

$$\psi_{j,k}(t) = \frac{1}{\sqrt{|s_0^j|}} \psi \left(\frac{t - k\tau_0 s_0^j}{s_0^j} \right) \quad (4)$$

em que j e k são inteiros; $s_0 > 1$ é um parâmetro de dilatação fixo; τ_0 é o fator de translação, que depende do fator de dilatação.

Geralmente escolhe-se $s_0 = 2$ para que se tenha uma amostragem da frequência chamada de

amostragem diádica e $\tau_0 = 1$ é escolhido para a amostragem temporal, também diádica. Isto resulta em (Oliveira, 2007):

$$\psi_{j,k}(t) = \sqrt{2^j} \psi(2^j t - k) \quad (5)$$

Quando são usadas *wavelets* discretas para analisar um sinal, o resultado é uma série de coeficientes *wavelet*, também chamados série de decomposição de *wavelet* (Oliveira, 2007). Como uma *wavelet* pode ser vista como um filtro passa-faixa, a série de *wavelets* escalonadas pode ser vista como um banco de filtros passa-faixa com fator Q (fator de fidelidade do banco de filtros). Na prática, tem-se uma *wavelet* discretizada, com limites superiores e inferiores para as translações e os escalonamentos, faltando apenas o cálculo de sua Transformada *Wavelet* Discreta (DWT) (Oliveira, 2007).

A ideia de passar um sinal por um banco de filtros é conhecida como codificação em sub-bandas (Subband Coding). Uma *Wavelet* discreta não é discreta no tempo, mas sim nas translações e escalonamentos. O sinal, ao passar por um banco de filtros digitais, resulta na DWT.



Fig. 3. Codificação em sub-bandas: H0 = filtro passa-baixa; H1 = filtro passa-alta; S(n) = entrada; c(n) e d(n) = saídas; $\downarrow 2$ = decimação por 2.

2.2.2 Filtragem:

Como filtros passa-baixas são utilizados na decomposição da imagem, a atenuação do ruído é um processo intrínseco à transformada, bem como a preservação de bordas.

2.2.3 Compressão:

O método de compressão de imagens baseado na transformada *wavelet* deve tirar vantagem da esparsidade das imagens geradas ao longo da transformada. Um levantamento da função de densidade de probabilidade dos coeficientes da imagem transformada mostra que esta distribuição é tipicamente Laplaciana, isto é, existe uma grande quantidade de valores muito próximos de zero, que cai exponencialmente conforme se desloca para as extremidades do gráfico da distribuição.

3 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNAs), são sistemas processadores de informação, inspirados na habilidade do cérebro humano de aprender a partir da observação e generalizar por abstração. O fato de as redes neurais artificiais serem capazes de aproximar relações de entrada / saída arbitrárias (Silva, 2002) tem levado a sua utilização para aplicações

totalmente diferentes, por exemplo: reconhecimento de padrões, processamento de voz e imagens, controle, aplicações médicas etc. A rede perceptron de múltiplas camadas (*Multilayer Perceptron - MLP*) consiste de um modelo sem realimentação formado por camadas de neurônios perceptron. O modelo de neurônio mais utilizado é baseado no trabalho de McCulloch e Pitts, ilustrado na Fig. 4(a).

Pela Fig. 4(a), o neurônio é dividido em duas partes: função da rede e função de ativação. A função da rede determina como as entradas da rede x_j ($1 \leq j \leq N$) são combinadas dentro do neurônio. Neste caso, uma combinação linear de pesos é utilizada, tal que:

$$u = \sum_{j=1}^N w_j x_j + \theta \quad (6)$$

em que w_j são os pesos sinápticos. A variável θ é chamada de bias e é usada para determinar o limiar de atuação do modelo.

A saída do neurônio, denotada por a na Fig. 4(a), está relacionada à entrada u através de uma transformação linear ou não-linear chamada de função de ativação (Haykin, 2001), como segue:

$$a = f(u) \quad (7)$$

Em vários modelos de redes neurais, diferentes funções de ativação têm sido propostas. A função de ativação utilizada por este projeto foi a tangente sigmóide:

$$u = \frac{1}{1 + e^{-\frac{u}{T_r}}} \quad (8)$$

Uma rede MLP típica é então apresentada na Fig. 4(b), em que cada nó representa um neurônio individual.

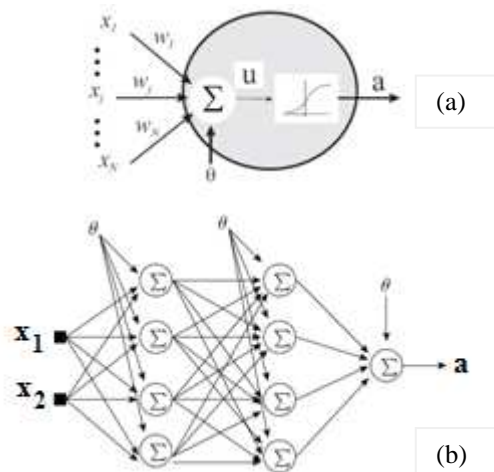


Fig. 4. (a) Neurônio perceptron ; (b) Configuração típica de uma rede perceptrons de múltiplas camadas.

Esses neurônios são organizados em camadas, denominadas primeira e segunda camadas ocultas e camada de saída. As entradas são apresentadas à rede através de uma pseudo-camada que não possui

nenhum modelo de neurônio implementado (Haykin, 2001). Desta forma, uma rede neural *multilayer perceptrons* é um processador paralelo distribuído, formado por unidades de processamento simples, que tem a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso.

3.1 Algoritmo Resilient Backpropagation (RPROP)

O algoritmo de treinamento RPROP é um eficiente esquema de aprendizagem que executa a adaptação direta da atualização dos pesos sinápticos baseada na informação do gradiente local. Um diferença crucial entre este e o algoritmo *Backpropagation* é o fato de que o esforço da adaptação dos pesos não é prejudicado pelo comportamento do gradiente, sendo introduzido um valor de atualização Δ_{ij} para cada peso sináptico, responsável por determinar apenas a amplitude da atualização do peso. Essa atualização adaptativa evolui durante o processo de treinamento, baseada na visão local da função custo E , de acordo com a seguinte regra de aprendizagem (Riedmiller, 1993).

$$\Delta_{ij}^{(t)} = \begin{cases} \eta^+ * \Delta_{ij}^{(t-1)}, & \text{se } \frac{\partial E^{(t-1)}}{\partial w_{ij}} * \frac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ \eta^- * \Delta_{ij}^{(t-1)}, & \text{se } \frac{\partial E^{(t-1)}}{\partial w_{ij}} * \frac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ \Delta_{ij}^{(t-1)}, & \text{caso contrário} \end{cases} \quad (9)$$

em que $0 < \eta^- < 1 < \eta^+ < t$ representa o número de épocas do treinamento. Seguindo a regra descrita em (9), toda vez que a derivada parcial correspondente ao peso w_{ij} muda seu sinal em relação ao passo de tempo anterior, quer dizer que a última atualização foi muito alta e o algoritmo passou por um mínimo local. Assim, o valor de atualização Δ_{ij} decresce pelo fator η^- . Caso a derivada mantenha o mesmo sinal, o valor de atualização é incrementado de forma a acelerar a convergência. Em resumo, se a derivada é positiva (aumentando o erro), o peso é reduzido pelo seu valor de atualização, caso a derivada seja negativa, o valor de atualização passa a ser então positivo (Riedmiller, 1993):

$$\Delta w_{ij}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)}, & \text{se } \frac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ +\Delta_{ij}^{(t)}, & \text{se } \frac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ 0, & \text{caso contrário} \end{cases} \quad (10)$$

No entanto, existe uma exceção: se a derivada parcial trocar de sinal, isto é, se o passo anterior for grande demais e o mínimo for ultrapassado, a atualização do peso, Δw_{ij} é revertida:

$$\Delta w_{ij}^{(t)} = -\Delta w_{ij}^{(t-1)} \text{ se } \frac{\partial E_{ij}^{(t-1)}}{\partial w_{ij}} * \frac{\partial E_{ij}^{(t)}}{\partial w_{ij}} < 0 \quad (11)$$

Por causa disso, a derivada deve mudar seu sinal novamente no passo seguinte, para evitar que ocorra uma punição novamente do valor de atualização. Não deve haver adaptação do valor de atualização no passo posterior. Uma forma prática de evitar isso é fazendo com que $\frac{\partial E_{ij}^{(t-1)}}{\partial w_{ij}} = 0$. Os valores de

atualização e os pesos somente são modificados depois que todo o conjunto de treinamento é apresentado à rede, o que caracteriza aprendizagem por lote ou batch (Riedmiller, 1993). Inicialmente, todos o valores de ajuste são iguais à constante Δ_0 , que é um dos parâmetros do RPROP. Se Δ_0 determina diretamente a amplitude do primeiro ajuste dos pesos, ele pode ser escolhido de acordo com a magnitude dos pesos iniciais, por exemplo, $\Delta_0 = 0,1$ (Riedmiller, 1993). A escolha desse valor não é crítica, já que seu valor é adaptado enquanto o treinamento ocorre (Silva, 2002).

No treinamento da rede através do algoritmo RPROP, para evitar uma variação excessiva dos pesos, define-se um parâmetro para o valor máximo de ajuste, $\Delta_{\text{máx}}$, que assume um valor igual a 50, sugerido por (Riedmiller, 1993). Os fatores de acréscimo e decréscimo são fixados em $\eta^+ = 1,2$ e $\eta^- = 0,5$. Estes valores são baseados em considerações teóricas e empíricas. Sendo assim, o número de parâmetros fica reduzido a dois, Δ_0 e $\Delta_{\text{máx}}$ (Silva, 2002; Riedmiller, 1993).

4 Sistema

O sistema descrito na Fig. 5 mostra todo o processo utilizado neste trabalho. Após a entrada dos dados (imagens), é preciso fazer um pré-processamento da informação. Ao se utilizar *wavelets*, existem diferentes possibilidades de escolha da função, denominada *wavelet* “mãe”, com características distintas e melhor adaptadas a determinados contextos. No entanto, neste trabalho foram desenvolvidos algoritmos para filtragem, baseados nas funções *wavelets*: Haar, Daubechies, Symlets, Cohen – Daubechies – Feauveau biorthogonal e Antonini – Barlaud – Mathieu – Daubechies, implementadas a partir de (Gonzalez, 2004). O algoritmo da transformada *wavelet* é feito da seguinte forma: o sinal é passado por dois filtros, um passa-baixas e um passa-altas, que irão dividir o sinal em outros dois sinais menores, sendo um de frequências maiores que um limiar (*threshold*) e outro de frequências menores. Este processo é repetido sobre o sinal de menores frequências até um nível predeterminado e os sinais de maior frequência são armazenados a cada iteração com os filtros. Estes sinais de maior frequência são conhecidos como coeficientes de *wavelet* (NPDI/DCC/UFMG, 2009).

Na Fig. 5, é possível visualizar todo o sistema desenvolvido neste trabalho. Inicialmente, houve a aquisição das imagens; posteriormente, a fase de pré-processamento, que pode utilizar as *wavelets* (Fig. 6)

ou não, para gerar os conjuntos de treinamento (Fig. 7) e testes (Fig. 8) que serão finalmente utilizados para treinamento e generalização da rede neural.

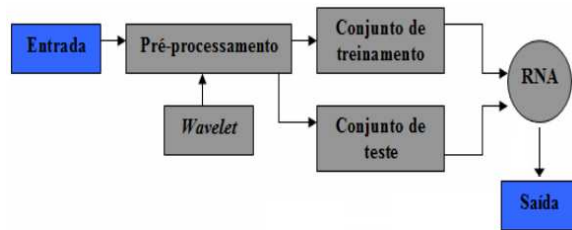


Fig. 5. Diagrama em blocos do sistema.

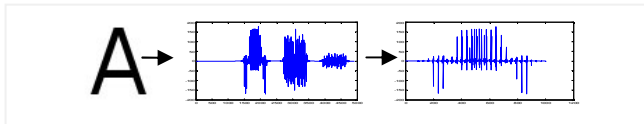


Fig. 6. Exemplo da Extração de Descritores Wavelets.



Fig. 7. Modelo do conjunto de treinamento.



Fig. 8. Modelo do conjunto de testes.

Na configuração da rede neural, foram utilizados os seguintes parâmetros:

- Rede: *Multilayer Perceptron*.
- Algoritmo: *Resilient Backpropagation* (RPROP).
- Condição de Parada: Nº de épocas (150) ou Erro ($10e-5$).
- Camadas: Entrada – Oculta – Oculta – Saída.

5 Resultados

Todas as simulações foram feitas com o software MATLAB 2008b de propriedade da *The MathWork*. O sistema operacional utilizado foi o GNU/Linux Ubuntu 9.04, instalado em um *hardware* PC *notebook*, Celeron M 1,6GHz, 2 GB RAM.

Os resultados obtidos estão dispostos na Tab. 01, de acordo com duas categorias de pré-processamento: sem *wavelet* - maior conjunto de treinamento e maior tempo para o aprendizado do mesmo pela rede; com *wavelet* - menor conjunto de treinamento, o que proporcionou o aprendizado da rede de maneira mais rápida.

É importante perceber que o critério de parada foi o número de épocas; o erro se manteve em torno de $10e-4$ e a generalização foi de 92%. Esta generalização foi capaz de reconhecer imagens das letras “A, E, I, O, U” com leves níveis de deslocamento 2D, em torno de 10% e ruído decorrelacionado (Fig. 8).

Tab. 1. Resultados: PP - Pré-processamento; CT - Conjunto de treinamento; TT -Tempo de treino; Erro; EP - Épocas; GE - generalização.

PP	CT	TT (min)	Erro (e-04)	GE
Sem Wavelet	4096x25	3.23	1.8340	92%
Com Wavelet	1000x25	1.43	7.7728	92%

Conclusão

A Rede Neural (MLP) foi utilizada eficientemente na classificação dos padrões apresentados, generalizando em ambos os casos com 92%.

O processo de extração de características foi simplificado e melhorado pelo uso da transformada *Wavelet* na fase de pré-processamento podendo assim otimizar principalmente o treinamento e a classificação da rede neural utilizada. Cabe ressaltar que não somente o método, mas também a sua capacidade de redução de dimensão, preservando o significado da informação, foi de extrema importância para diminuir o custo computacional.

Finalmente, é importante lembrar que o objetivo principal do trabalho não é construir um classificador ótimo, mas sim, mostrar possibilidades que possam otimizar esse processo. Por isso, optou-se por utilizar um pequeno conjunto de treinamento e teste, que mesmo sendo reduzidos e não muito diversificados, ainda servem como base para projetos maiores.

Referências

- Castleman K. R. (1996), “Digital Imaging Processing”, 1ª Ed. Prentice Hall.
- Gonzalez R. C., Woods R. E., Eddins S. L., (2004). “Digital Image Processing Using Matlab”, 1ª Ed. Prentice Hall.
- Haykin S. (2001). Redes Neurais – princípios e prática. Bookman.
- Mallat S. (1999), “A Wavelet Tour of Signal Processing”, 2ª Ed. Academic Press.
- NPDI/DCC/UFGM. (2009). Artigo : Extração de características utilizando wavelets para indexação de Imagens, obtido através do endereço: <http://wavelet.dcc.ufmg.br/npdi/modules/news3/article.php?storyid=117>.
- Oliveira H. M. (2007). Análise de fourier e wavelets: sinais estacionários e não estacionários. 1ª Ed. Ed. Universitária UFPE.
- Riedmiller M. and Braun H. (1993). A direct adaptative method for faster backpropagation learning: the RPROP algorithm’, Proceedings of the IEEE International Conference on Neural Networks, San Francisco, EUA, 586–591.
- Silva P. H da F. (2002). Modelos através de redes neurais artificiais sem realimentação para dispositivos e circuitos de RF/microondas. Tese de doutorado, Universidade Federal da Paraíba.