

Predição Multiescala de Tráfego de Redes Utilizando Redes Neurais RBF Treinadas com Algoritmo de Mínimos Quadrados Ortogonais

F.H.T. VIEIRA¹, R.P. LEMOS², Escola de Engenharia Elétrica, Universidade Federal de Goiás - UFG, Av. Universitária, 1488, Quadra 86, Bloco A, Setor Leste Universitário, 74605-010 Goiânia, GO, Brasil.

L.L. LING³, Departamento de Comunicações, FEEC, Universidade Estadual de Campinas - UNICAMP, Cx.P. 6110, 13083-970 Campinas, SP, Brasil.

Resumo. Neste artigo, apresentamos um novo algoritmo de treinamento para redes neurais RBF (Função de Base Radial) baseado em mínimos quadrados ortogonais e em decomposição multiescala de sinais. Propomos um algoritmo de predição de séries temporais que combina as predições de aproximações para estas mesmas séries e de seus detalhes em diferentes escalas através da Transformada Wavelet. Aplicamos as redes neurais RBF treinadas com o algoritmo proposto na predição de tráfego de redes de computadores. O treinamento das redes neurais RBF com algoritmo de mínimos quadrados ortogonais aliada à decomposição wavelet contribui para evitar problemas de mal condicionamento da matriz de interpolação, como também para melhorar a capacidade de extrapolação da rede neural RBF. Esta última característica é verificada pela redução do erro quadrático médio de predição. As simulações realizadas confirmam que predições mais precisas são obtidas para as séries temporais de tráfego de redes em relação a outras redes neurais existentes.

Palavras-chave. Predição, Rede Neural, Tráfego de Redes, Mínimos Quadrados Ortogonais.

1. Introdução

As redes neurais MLP (*Multilayer Perceptron*) e RBF (*Radial Basis Function*) constituem a espinha dorsal das redes neurais com aprendizagem supervisionada. As redes neurais podem ser usadas em tomadas de decisões complexas e para aproximar a dinâmica de certos sistemas. A rede neural RBF em particular, tem sido aplicada com sucesso na predição de séries temporais não-lineares [9].

Neste artigo, desenvolvemos um treinamento para redes neurais RBF baseado em mínimos quadrados ortogonais (OLS- *Orthogonal Least Squares*) e em decomposição

¹flavio@eee.ufg.br

²lemons@eee.ufg.br

³lee@decom.fee.unicamp.br

wavelet. Note que o algoritmo de mínimos quadrados ortogonais tem sido usado com eficiência em outras aplicações como em sistemas de controle [2], redes neurais nebulosas [6], etc. Em seguida, utilizamos a rede neural RBF como ferramenta para prever o comportamento do tráfego em redes de computadores a fim de propiciar aos mecanismos de controle de tráfego informações seguras para tomada de decisões. Mais especificamente, analisamos o problema da predição de valores futuros de uma série temporal correspondente ao tráfego de redes.

A maioria dos métodos de predição existentes efetuam uma análise global dos dados. Tais métodos levam à perda de detalhes (*underfitting*) ou a inclusão de ruído (*overfitting*), por tratar os dados de forma única. Além do mais, uma vez que sinais reais podem ser não apenas não-lineares, mas também não-estacionários; é necessário desenvolver preditores mais robustos e que lidem com essas características. Mostraremos que um treinamento robusto pode ser obtido ao se considerar os detalhes das séries temporais em diferentes escalas e o mesmo sinal com uma menor riqueza de detalhes. Para isso, propomos um arquitetura neural constituída por redes neurais RBF treinadas com o algoritmo de mínimos quadrados ortogonais, onde diferentes redes neurais RBF ficam responsáveis pela predição dos detalhes e da aproximação da série de tráfego em questão.

2. A Rede Neural RBF

A rede neural RBF em geral possui uma camada de nós de entrada (nós sensoriais), uma camada de nós escondidos e uma camada de saída. Nesta rede neural, a ativação de um nó é função da distância entre seus vetores de entrada e os pesos, daí o nome Funções de Base Radial. Na saída da rede RBF, os parâmetros ajustáveis são os pesos de uma combinação linear. As redes RBF constroem aproximações locais para mapeamentos entrada-saída, resultando em sensibilidade reduzida à ordem de apresentação dos dados do conjunto de treinamento [7][1].

Seja um número inteiro $P < N$ representando o número de neurônios da camada intermediária e N é igual ao número de dados de treinamento. A seguinte equação fornece a saída da rede neural da Figura 1

$$y = \sum_{k=0}^P w_k \varphi(\underline{x}; t_k) + w_o, \quad (2.1)$$

onde t_k representa o centro da função de base radial $\varphi(\underline{x}; t_k)$ e w_k é o peso da rede neural RBF para a função de base radial k .

Os pesos w_k da rede neural RBF podem ser encontrados através do método de mínimos quadrados. Ou seja, deve-se encontrar os pesos w_k de forma a minimizar a função custo ε_R dada por

$$\varepsilon_R = \sum_{i=1}^N (d_i - f(x_i))^2, \quad (2.2)$$

onde $\underline{d} = [d_1, d_2, \dots, d_N]^T$ é o vetor de respostas desejadas, $\underline{w} = [w_0, w_1, w_2, \dots, w_P]^T$ é o vetor de pesos da rede neural RBF e $f(x_i)$ é a saída da rede neural para o

vetor de entrada x_i . Além do cálculo dos pesos, o treinamento da rede neural RBF consiste em determinar as localizações dos centros \underline{t}_k das funções de base radial dos P nós de sua camada escondida.

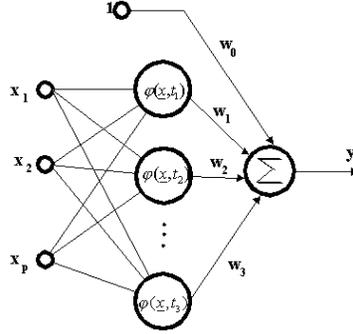


Figura 1: Rede neural RBF.

Investigações experimentais e teóricas sugerem que a escolha do tipo de não-linearidade para a função de base radial não é crucial para o desempenho da rede neural [10]. Assumindo que as funções de base radial sejam gaussianas, a saída da rede neural RBF pode ser escrita como

$$y = \sum_{k=0}^P w_k \exp\left(-\frac{1}{\sigma_k^2} \|\underline{x} - \underline{t}_k\|^2\right) + w_o, \quad (2.3)$$

onde σ_k representa a largura da função radial e \underline{t}_k o seu centro.

Seja Φ uma matriz de interpolação de tamanho $N \times (P + 1)$, dada por

$$\Phi = \begin{bmatrix} 1 & \varphi(\underline{x}_1, \underline{t}_1) & \varphi(\underline{x}_1, \underline{t}_2) & \cdots & \varphi(\underline{x}_1, \underline{t}_P) \\ 1 & \varphi(\underline{x}_2, \underline{t}_1) & \varphi(\underline{x}_2, \underline{t}_2) & \cdots & \varphi(\underline{x}_2, \underline{t}_P) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \varphi(\underline{x}_N, \underline{t}_1) & \varphi(\underline{x}_N, \underline{t}_2) & \cdots & \varphi(\underline{x}_N, \underline{t}_P) \end{bmatrix}.$$

Em termos matriciais, o vetor de pesos \underline{w} que minimiza a função custo ε_R (equação 2.2) é dado em função da matriz de interpolação Φ da seguinte forma [4]

$$\underline{w} = (\Phi^T \cdot \Phi)^{-1} \Phi^T \cdot \underline{d}. \quad (2.4)$$

Mal condicionamento da matriz de interpolação Φ pode ocorrer pela escolha de centros próximos. O algoritmo de mínimos quadrados acrescido de uma ortogonalização dos vetores de entrada de dados pode ser usado para seleção de um conjunto de centros a partir de candidatos fixos e evitar problemas de mal condicionamento, além de obter redes com um menor número de centros.

3. Mínimos Quadrados Ortogonais

As redes neurais MLPs possuem aprendizagens baseadas em técnicas de otimização não-linear, podendo apresentar problemas de mínimos locais [4]. Outras técnicas de otimização aplicáveis ao treinamento de redes neurais como os algoritmos genéticos requerem um longo tempo de processamento.

A seleção direta é um algoritmo não-linear que procura em um espaço discreto de conjuntos um subconjunto com o menor erro de predição [8]. Inicia-se com um subconjunto vazio e acrescenta-se a cada iteração uma função de base (centro) de tal forma a reduzir a soma dos erros quadráticos. Esse processo se repete até que um determinado critério seja atendido.

Nesta seção, apresentamos o algoritmo de mínimos quadrados ortogonais como um modo de seleção direta dos centros das funções de base radial. Para tal, fazemos uso da ortogonalização de Gram-Schmidt que assegura que cada nova coluna a ser acrescentada na matriz de projeto (interpolação) seja perpendicular a todas as outras colunas [4].

Denotaremos a matriz de projeto H por

$$H = \begin{bmatrix} h_1(x_1) & h_2(x_1) & \cdots & h_m(x_1) \\ h_1(x_2) & h_2(x_2) & \cdots & h_m(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(x_p) & h_2(x_p) & \cdots & h_m(x_p) \end{bmatrix}, \quad (3.1)$$

onde cada coluna desta matriz corresponde a uma função de base radial a ser escolhida. Para reduzir a complexidade computacional da seleção direta, aplicaremos a ortogonalização de Gram-Schmidt. Assim, a matriz de projeto H é fatorada em $H_m = \tilde{H}_m U_m$, em que \tilde{H}_m possui colunas ortogonais entre si e U_m é uma matriz triangular superior.

Seja $Z = [z_1 \ z_2 \ \cdots \ z_M]$ uma matriz cujas colunas $\{z_f\}_{f=1}^M$ correspondem aos M possíveis vetores candidatos a funções de base dos neurônios da rede neural RBF. A cada passo m , acrescenta-se à matriz de projeto ortogonalizada $\tilde{H}_m = [H_{m-1} \ \tilde{z}_i]$, um vetor \tilde{z}_i calculado pela seguinte equação

$$\tilde{z}_f = z_f - \sum_{j=1}^m \frac{z_f^T \tilde{h}_j}{\tilde{h}_j^T \tilde{h}_j} \tilde{h}_j. \quad (3.2)$$

Pode-se demonstrar que essa escolha para \tilde{z}_f resulta em uma maior diminuição do erro quadrático médio (função custo ε_R), maximizando a seguinte equação (ver Apêndice)

$$\varepsilon_{R_m} - \varepsilon_{R_{m+1}} = \frac{(\underline{d}^T \tilde{z}_f)^2}{\tilde{z}_f^T \tilde{z}_f}. \quad (3.3)$$

Uma vez que a matriz de projeto é $H_m = \tilde{H}_m U_m$, o vetor de pesos pode ser escrito como

$$\underline{w}_m = U_m^{-1} \tilde{\underline{w}}_m, \quad (3.4)$$

onde

$$\tilde{w}_m = \frac{d^T \tilde{h}_j}{\tilde{h}_j^T \tilde{h}_j}, \quad (3.5)$$

em que a matriz triangular superior U_m é dada por

$$U_m = \begin{bmatrix} U_{m-1} & (\tilde{H}_{m-1}^T \tilde{H}_{m-1})^{-1} \tilde{H}_{m-1}^T z_f \\ 0_{m-1}^T & 1 \end{bmatrix}. \quad (3.6)$$

A decisão de se interromper o acréscimo de funções base pode ser feita monitorando-se algum critério de seleção. Um dos critérios mais utilizados é o critério de validação cruzada generalizado, dado por [8]

$$\sigma_m^2 = \frac{n d^T P_m^2 d}{(\text{traço}(P_m))^2}, \quad (3.7)$$

onde n é o número de padrões de treinamento e P é a matriz de projeção do erro. A matriz de projeção P_m é atualizada através da seguinte equação (ver Apêndice)

$$P_m = I_n - \sum_{j=1}^m \frac{\tilde{h}_j \tilde{h}_j^T}{\tilde{h}_j^T \tilde{h}_j}. \quad (3.8)$$

Interrompe-se o acréscimo de funções de base radial no momento em que o valor de σ_m^2 pára de diminuir e começa a aumentar.

4. Treinamento Multiescala através da Transformada Wavelet

A transformada wavelet proporciona uma representação multiescala do sinal. Uma função wavelet φ tem suporte compacto, ou seja, não tem componentes espectrais fora de um faixa de frequências e é oscilatória [14]. Por dilatações e translações desta função wavelet mãe φ obtemos uma família de funções através da equação

$$\varphi_{s,t}(u) = |s|^{-p} \varphi\left(\frac{u-t}{s}\right), \quad (4.1)$$

onde $s \in \mathbb{R}$, $p \geq 0$, s é o parâmetro de dilatação e t é o parâmetro de translação. A Transformada Wavelet Contínua (CWT) de uma função $f(t)$ é definida como

$$W(s,t) = \int_{-\infty}^{+\infty} f(u) \varphi_{s,t}(u) du = \langle f, \varphi_{s,t} \rangle. \quad (4.2)$$

Obtém-se a transformada wavelet discreta pela discretização temporal e em escala da transformada contínua. Utilizando uma discretização diádica em que $s = 2^j$, $t = k2^j t_0$ e $j, k \in \mathbb{Z}$, a wavelet mãe $\varphi_{j,k}$ (4.1) pode ser escrita como [14]

$$\varphi_{j,k}(u) = |2|^{-\frac{j}{2}} \varphi(2^{-j}u - k). \quad (4.3)$$

Seja uma função $\phi \in L^2(\mathbb{R})$, que será denominada de função de escala. A família de funções

$$\phi_{j,k}(u) = |2|^{-\frac{j}{2}} \phi(2^{-j}u - k), \quad j, k \in \mathbb{Z} \quad (4.4)$$

é uma base ortonormal do subespaço V_j chamado de espaço de escala. O espaço de escala V_j é formado pelas funções cujos detalhes estão na escala 2^j . Uma função $f \in L^2(\mathbb{R})$ pode ser representada por projeção ortogonal em V_j [14]

$$P_{V_j}(f) = \sum_k \langle f, \phi_{j,k} \rangle \phi_{j,k}. \quad (4.5)$$

O espaço V_{j-1} é obtido acrescentando-se a V_j todas as funções de $L^2(\mathbb{R})$ com frequências na faixa $[\alpha_j, \alpha_{j-1}]$ relativas ao espaço vetorial W_j . O espaço de detalhes W_j é gerado pela base ortonormal wavelet $\{\varphi_{j,k}, k \in \mathbb{Z}\}$. Assim, um sinal $f(x)$ pode ser representado em função de sua aproximação em uma escala G acrescido de seus detalhes, ou seja [14]

$$f(x) = P_{V_G}(f(x)) + \sum_{j=1}^G P_{W_j}(f(x)). \quad (4.6)$$

Note que o processo $f(x)$ é reconstruído de maneira aditiva através da equação (4.6). De forma análoga, propomos adicionar as predições das redes neurais RBF para obter a predição de h passos a frente da série original, ou seja

$$f(x+h) \cong \hat{P}_{V_G}(f(x+h)) + \sum_{j=1}^G \hat{P}_{W_j}(f(x+h)), \quad (4.7)$$

onde \hat{P} representa a estimativa da rede neural RBF correspondente a equação (2.1). Em outras palavras, podemos aplicar a transformada wavelet na série temporal a ser predita obtendo uma série aproximada e os processos de detalhes em várias escalas. Assim, treinamos a rede RBF para prever amostras futuras da série aproximada e das séries de detalhes. A predição final como mostra a equação (4.7) será a soma das predições das amostras da série aproximada mais as predições das amostras das séries de detalhes realizadas pela rede RBF.

5. Simulações e Resultados

Nesta seção, verificamos o desempenho do treinamento multiescala proposto aplicado a redes neurais RBF com relação a predição de amostras futuras de intensidade de tráfego de redes. As séries temporais utilizadas neste trabalho foram retiradas das medições de tráfego da Bellcore, como por exemplo os arquivos BC-Octint e BC-OctExt ⁴, os quais são utilizados em diversos estudos [12].

⁴<http://ita.ee.lbl.gov/html/traces.html>

Para avaliar o desempenho do algoritmo de predição proposto foi utilizado o erro quadrático médio normalizado EQMN dado por

$$EQMN = \frac{1}{\sigma^2 p} \sum_{k=1}^p [y(k) - \hat{y}(k)]^2, \quad (5.1)$$

onde $y(k)$ é o valor real da série, $\hat{y}(k)$ é o valor predito, σ^2 é a variância da seqüência real sobre o intervalo de duração da predição e p é o número de amostras do teste.

Primeiramente, analisamos a capacidade de predição de um instante a frente do algoritmo de predição proposto para a série Bc-Octext. A predição de amostras futuras desta série foi realizada utilizando redes neurais RBF com cinco amostras de dados como entrada para a rede neural. Estas amostras de entrada correspondem a valores consecutivos do traço de tráfego considerado. As saídas desejadas utilizadas para o treinamento da rede RBF são os valores a um passo a frente da série Bc-Octext. Além disso, considerou-se a decomposição wavelet desta série até a escala $j = 3$. Neste caso, as redes neurais RBF foram treinadas com 1000 exemplos de treinamento da série Bc-Octext e os 1000 pontos restantes da mesma foram usados no teste de predição. O EQMN total obtido para a série Bc-Octext foi 0,2068, que representa um baixo valor comparado a outros algoritmos. A Tabela 1 compara os EQMNs obtidos para essa mesma série com outros preditores neurais e ainda redes neurais RBF treinadas somente com algoritmo de mínimos quadrados ortogonais.

A série de tráfego Bc-Octint é um bom exemplo de série não-estacionária. Esta série temporal apresenta um salto estatístico em torno do instante 1100, oferecendo maior dificuldade de predição para as redes neurais. Na simulação para a série Bc-Octint utilizamos 30 elementos (amostras) para o vetor de entrada com 900 pontos de treinamento e 800 pontos de teste. A Figura 2 mostra as predições a um passo (1s a frente) realizadas para a série Bc-octint. Para esta série temporal, obteve-se um EQMN total de 0,1331 pela decomposição do processo de tráfego novamente até a escala 3.

O algoritmo proposto atingiu tal resultado devido entre outros fatores, à capacidade de predição da rede RBF de variações bruscas “retangulares” que surgem com as aproximações do sinal com menos detalhes. A rede neural FIR MLP por exemplo, encontrou maior dificuldade para predizer variações deste tipo [13].

Comparando-se o desempenho de predição de outras redes neurais, nota-se uma redução significativa do EQMN de predição (Tabela 1) em relação às redes neurais RBF treinadas com o algoritmo proposto. Foram utilizados na comparação o mesmo número de pontos de treinamento e teste para as redes neurais RBF, FIR MLP e MLP.

Mantendo o mesmo número de entradas das redes neurais, verificamos a predição da série Bc-Octint utilizando diferentes níveis de detalhes (experimentos 1, 2, 3 e 4) como mostra a Tabela 2. Este teste revela que há uma diminuição do EQMN a medida que detalhes em escalas menores vão sendo considerados. Nota-se também um melhor desempenho de predição pela rede neural RBF das aproximações com menos detalhes.

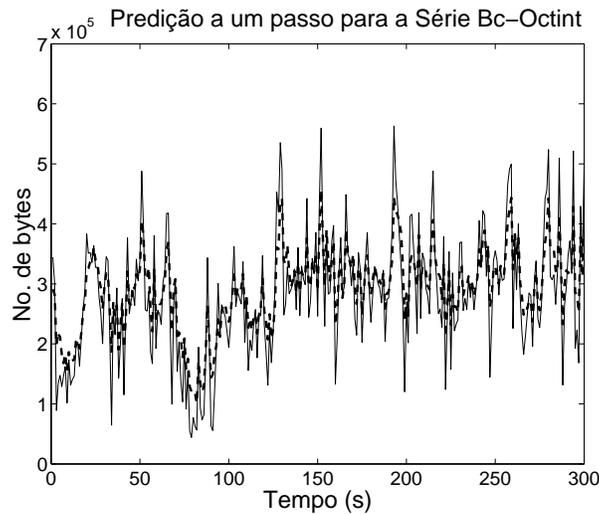


Figura 2: Predição (tracejado) da série Bc-Octint. EQMN=0,1331.

Tabela 1: EQMN de predição de redes neurais

EQMN				
Série	MLP	FIR MLP	RBF+OLS	RBF+OLS+Wavelets
Bc-Octext	0,4037	0,4260	0,3962	0,2068
Bc-Octint	1,21	0,7408	0,5092	0,1331

6. Conclusões

Neste artigo, empregamos o conhecimento dos detalhes das séries de tráfego em diversas escalas em conjunto com o algoritmo de mínimos quadrados ortogonais, trazendo uma melhoria significativa no desempenho de predição.

Mostrou-se que o algoritmo proposto é uma ferramenta eficiente na modelagem e predição do comportamento de sistemas complexos, independentemente da escolha inicial dos pesos da rede neural. O algoritmo de mínimos quadrados ortogonais evita que a matriz de interpolação seja mal condicionada uma vez que são obtidos vetores ortogonais para esta matriz. Esse algoritmo também reduz o número de centros necessários em relação ao treinamento com mínimos quadrados ordinários. Além disso, o método de predição proposto também possibilita o uso de várias redes neurais na predição das séries que irão compor a série original. Pode-se então, atingir o limite máximo de desempenho de predição [11] para uma série temporal

Tabela 2: EQMN dos experimentos 1, 2, 3 e 4

EQMN						
	Aprox1	Det1	Det2	Det3	Det4	EQMN
Exp1	0,2301	0,6058	-	-	-	0,2552
Exp2	0,0875	0,6058	0,3262	-	-	0,1657
Exp3	0,0294	0,6058	0,3262	0,1896	-	0,1331
Exp4	0,0185	0,6058	0,3262	0,1896	0,1896	0,1319

pelo custo computacional de inclusão de redes neurais RBF em sua predição.

Apêndice

O vetor de erro \underline{e} entre o valor desejado e a resposta da rede neural \underline{f} é dado por

$$\underline{e} = \underline{d} - \underline{f} = \underline{d} - (I_p - HA^{-1}H^T)\underline{d} = P\underline{d}, \tag{6.1}$$

onde $A = H^T H$ e $P = (I_n - HA^{-1}H^T)$. Usando o fato de que $H_m = \tilde{H}_m U_m$ na matriz de projeção $P = (I_n - HA^{-1}H^T)$, tem-se que

$$P_m = (I_n - \tilde{H}_m(\tilde{H}_m^T \tilde{H}_m)^{-1})\tilde{H}_m^T, \tag{6.2}$$

$$P_m = I_n - \sum_{j=1}^m \frac{\tilde{h}_j \tilde{h}_j^T}{\tilde{h}_j^T \tilde{h}_j}.$$

A função custo ε_R (equação (2.2)) pode ser dada em função de P e \underline{d} da seguinte forma

$$\varepsilon_R = (\underline{d} - \underline{f})^T (\underline{d} - \underline{f}) = \underline{d}^T P^2 \underline{d} = \underline{d}^T P \underline{d}.$$

Assim, a variação da soma de erros quadráticos $\varepsilon_{R_m} - \varepsilon_{R_{m+1}}$ pode ser expressa por

$$\varepsilon_{R_m} - \varepsilon_{R_{m+1}} = \underline{d}^T (P_m - P_{m+1})\underline{d} = \frac{(\underline{d}^T \tilde{z}_f)^2}{\tilde{z}_f^T \tilde{z}_f}. \tag{6.3}$$

Abstract. In this paper, we present a novel training algorithm for RBF (Radial Basis Function) neural networks that is based both on multiscale decomposition of signals and the orthogonal least squares method. We propose a prediction algorithm of time series that combines the signal approximations and details predictions in different resolution levels through the Wavelet Transform. The proposed training algorithm contributes to avoid problems of ill-conditioning as well as to give an extrapolation capacity improvement verified by the mean-square error reduction of the prediction. We confirm through simulations that more precise predictions are obtained for the traffic traces in relation to other existing neural networks.

Referências

- [1] E. Avci, An expert system based on Wavelet Neural Network-Adaptive Norm Entropy for scale invariant texture classification, *Expert Systems with Applications* **32**, No. 3, (2007), 919-926.
- [2] S. Chen, S.A. Billings, Neural networks for nonlinear dynamic system modeling and identification, *International Journal of Control* **56**, No. 2 (1992), 319-346.
- [3] G. Cybenko, Approximations by superposition of a sigmoidal function, *Math. Control Signal Systems* **2**, No. 4 (1989), 303-314.
- [4] S. Haykin, “Neural Networks - A Comprehensive Foundation”, Prentice Hall, 2ed., 1998.
- [5] X. He, A. Lapedes, “Nonlinear Modeling and Prediction by Successive Approximations using Radial Basis Functions”, Technical Report, Los Alamos National Laboratory, 1991.
- [6] J.-S.R. Jang, C.-T. Sun, Functional equivalence between radial basis function networks and fuzzy inference systems, *IEEE Transactions on Neural Networks*, **4** (1993), 156-159.
- [7] W.W.Y. Ng, A. Dorado, D.S. Yeung, W. Pedrycz, E. Izquierdo, Image classification with the use of radial basis function neural networks and the minimization of the localized generalization error. *Pattern Recognition* **40**, No. 1 (2007), 19-32.
- [8] M.J.L. Orr, Regularization in the selection of radial basis function centres, *Neural Computation*, **7**, No. 3 (1995), 606-623.
- [9] M.A.S. Potts, D.S. Broomhead, Time series prediction with a radial basis function neural network, *SPIE Adaptive Signal Processing*, **1565** (1991), 255-266.
- [10] M.J.D. Powell. Radial basis function approximations to polynomials, in “Proc. 12th Biennial Numerical Analysis Conf.” (Dundee), pp. 223-241, 1987.
- [11] A. Sang, S.Q. Li, A predictability analysis of network traffic. in “Conference on Computer Communications”, *IEEE Infocom*, New York, Mar. 2000.
- [12] D. Veitch, P. Abry, A wavelet based joint estimator of the parameters of long-range dependence, *IEEE Trans. Inform. Theory-Special Issue on Multiscale Statistical Signal Analysis and Its Applications* **45**, No. 3 (1999), 878-897.
- [13] F.H.T. Vieira, “Predição de Tráfego em Redes de Comunicações utilizando Redes Neurais e Análise Wavelet - Alocação Dinâmica de Largura de Faixa”. Dissertação de mestrado. Universidade Federal de Goiás, Goiânia, Goiás, Brasil, 2002.
- [14] D.F. Walnut, “An Introduction to Wavelet Analysis”. Birkhäuser Boston, 1ed., 2004.