

# Algoritmos Heurísticos para a Seleção de Neurônios em Redes Neurais Polinômios Potências de Sigmóide (PPS) - Wavelet<sup>1</sup>

R.A.B. de QUEIROZ<sup>2</sup>, Departamento de Matemática Aplicada e Estatística, Laboratório de Computação de Alto Desempenho (LCAD), Instituto de Ciências Matemáticas e de Computação, USP, São Carlos, SP, Brasil

J.F. MARAR<sup>3</sup>, Departamento de Computação, Laboratório Sistemas Adaptativos e Computação Inteligente (Lab. SACI), Faculdade de Ciências, UNESP, Bauru, SP, Brasil.

**Resumo.** Este artigo trata-se da aplicação das funções PPS-wavelet na área de redes neurais artificiais. Subdividimos estes estudos em duas partes, na primeira parte, serão apresentados os aspectos fundamentais das funções PPS e uma aplicação da utilização das funções PPS-wavelets para o desenvolvimento de redes neurais artificiais com funções de ativação wavelets e, a seguir, serão mostrados três algoritmos para a seleção das funções de ativação, em ordem de apresentação, a técnica de seleção de funções baseada em valor de resíduo, a técnica de seleção de funções passo-a-passo por ortogonalização e a técnica de eliminação recorrente de funções. Todas estas técnicas foram investigadas e implementadas para as redes neurais PPS-wavelet. Um dos principais objetivos da implementação destas técnicas é a contribuição que elas trazem no contexto de inicialização dos parâmetros das redes neurais PPS-wavelets, que são as dilatações, as translações e os pesos sinápticos; de maneira a determinar, segundo cada técnica utilizada, os “melhores” neurônios para compor a camada escondida da arquitetura neural em estudo.

**Palavras-chave.** Polinômios potências de sigmóide, Redes neurais artificiais, Seleção de neurônios.

## 1. Introdução

Redes Neurais Artificiais (RNAs) são modelos biologicamente inspirados suportados por três importantes estruturas matemáticas: o neurônio, a arquitetura de interconexão dos neurônios e o algoritmo de aprendizagem. Para os propósitos desta

---

<sup>1</sup>Apoio financeiro parcial do PIBIC-CNPq (Processo 108250/2005-9) e da CAPES (Processo 3634/06-0)

<sup>2</sup>bonfim@icmc.usp.br, mestrando ICMC-USP

<sup>3</sup>fermarar@fc.unesp.br, Coordenador do Laboratório Sistemas Adaptativos e Computação Inteligente

pesquisa, serão adotadas como funções de transferência, ou de ativação, do neurônio as funções PPS-wavelets [9], a arquitetura de interconexão corresponde a topologia *feedforward* [5] e o algoritmo de treinamento é baseado no método gradiente descendente [8, 9]. A escolha arbitrária do conjunto de neurônios dispostos na arquitetura neural poderá influenciar o grau de complexidade do treinamento. Desta forma, é importante no processo de inicialização de uma RNA adotar o uso de técnicas para selecionar quais neurônios atuam de melhor maneira junto aos dados de entrada do problema [7, 11, 14]. O processo de treinamento da rede neural é um problema de otimização difícil, pois na matemática ainda não temos um método que dê garantia de encontrar o mínimo global para problemas dessa natureza. Neste sentido, investigamos a inicialização de uma RNA PPS-wavelet através de técnicas para seleção de neurônios da rede, em destaque, as técnicas seleção de funções baseada em valor de resíduo [1, 15], seleção de funções passo-a-passo por ortogonalização [1, 15] e eliminação recorrente de funções [1, 15] foram estudadas e implementadas. O uso dessas técnicas tem como objetivo auxiliar no processo de inicialização dos parâmetros da RNA PPS-wavelet, que são as dilatações, as translações e os pesos sinápticos; de maneira a determinar os “melhores” neurônios, segundo cada técnica utilizada. Possibilitando assim, a composição da camada escondida da arquitetura neural em estudo. O artigo encontra-se organizado da seguinte forma: na seção 2 apresentam-se os fundamentos teóricos das funções PPS, na seção 3 mostram-se as redes neurais projetadas através de funções PPS-wavelets, na seção 4 apresentam-se os algoritmos de seleção de neurônios e por fim as conclusões.

## 2. Teoria das funções PPS

As funções PPS correspondem a uma nova ferramenta matemática para problemas de aproximação de funções [10]; introduzidas por Marar em 1996 [8]. Dentre suas aplicações, temos: aproximação de funções [4, 6, 9, 12], redes neurais artificiais [9, 12, 13] e a construção de uma família de funções wavelets polinomiais [8]. De maneira a investigar os novos rumos científicos desta família de funções para construção de arquiteturas neurais PPS-wavelet, a seguir, vamos apresentar os aspectos teóricos que contemplam a teoria das funções PPS.

### 2.1. Funções sigmóides

Sigmóide corresponde a uma classe de funções dotadas das seguintes propriedades: monotonicamente crescente, diferenciável, limitada e cujo o gráfico lembra a forma do carácter “S”. As funções sigmóides, em sua grande maioria, satisfazem a condição

$$\Upsilon(x) = \begin{cases} 1, & x \rightarrow +\infty, \\ 0, & x \rightarrow -\infty. \end{cases}$$

## 2.2. Funções potências de sigmóide

Seja  $\Upsilon : \mathbf{R} \rightarrow [0,1]$  a função sigmóide definida por

$$\Upsilon(x) = \frac{1}{1 + e^{-x}}. \quad (2.1)$$

Chama-se função potência de sigmóide de expoente  $n$  ( $n \in \mathbf{Z}_+$ ) da  $\Upsilon$  a função  $\Upsilon^n : \mathbf{R} \rightarrow [0,1]$  definida por

$$\Upsilon^n(x) = \left( \frac{1}{1 + e^{-x}} \right)^n.$$

O conjunto das funções potências de sigmóide é definido por

$$\Phi = \{\Upsilon^0(x), \Upsilon^1(x), \Upsilon^2(x), \dots, \Upsilon^n(x), \dots\},$$

onde

$$\Upsilon^0(x) = \left( \frac{1}{1 + e^{-x}} \right)^0 = 1;$$

$$\Upsilon^1(x) = \left( \frac{1}{1 + e^{-x}} \right)^1 = \frac{1}{1 + e^{-x}};$$

$$\Upsilon^2(x) = \Upsilon^1(x) \cdot \Upsilon^1(x) = \frac{1}{1 + 2e^{-x} + e^{-2x}}.$$

A potência  $n$ -ésima da função sigmóide é representada por

$$\Upsilon^n(x) = \frac{1}{a_0 + a_1 e^{-x} + a_2 e^{-2x} + \dots + a_n e^{-nx}},$$

nos quais os coeficientes  $a_0, a_1, a_2, \dots, a_n$  podem ser obtidos da linha  $n$ -ésima do triângulo de Pascal [9]. As funções PPS são definidas como combinações lineares das funções potências de sigmóide.

## 2.3. Derivada da função sigmóide

A derivada da função sigmóide é calculada da seguinte forma

$$\begin{aligned} \frac{d[\Upsilon(x)]}{dx} &= \frac{d\left[\frac{1}{1+e^{-x}}\right]}{dx} = \frac{e^{-x}}{(1+e^{-x})^2} \\ &= \frac{e^{-x}}{(1+e^{-x}) \cdot (1+e^{-x})} = \Upsilon(x) \left( \frac{e^{-x}}{1+e^{-x}} \right) \\ &= \Upsilon(x) \left( \frac{e^{-x} + 1 - 1}{1+e^{-x}} \right) = \Upsilon(x) \left( \frac{1+e^{-x}}{1+e^{-x}} - \frac{1}{1+e^{-x}} \right) \\ &= \Upsilon(x) (1 - \Upsilon(x)) = -\Upsilon^2(x) + \Upsilon(x) \end{aligned}$$

A derivada primeira da função sigmóide é definida como função PPS-Radial [9].

## 2.4. Família de funções PPS-wavelet

O processo de derivação sucessiva da função sigmóide definida na Eq. (2.1) possibilita a construção de uma família de funções wavelets polinomiais muito útil para a construção de redes neurais wavelets (ou wavenet, que é apenas uma contração das palavras redes neurais wavelets)[9]. O primeiro elemento desta família é definido por

$$\varphi_2(x) = 2\Upsilon^3(x) - 3\Upsilon^2(x) + \Upsilon(x),$$

onde  $\varphi_2(x)$  é o resultado da derivada segunda da função sigmóide definida na Eq. (2.1). Abaixo, os resultados correspondem as funções analíticas para os elementos  $\varphi_3(x)$ ,  $\varphi_4(x)$ ,  $\varphi_5(x)$  que são ilustrados na Figura 1 e representados, respectivamente, por

$$\begin{aligned}\varphi_3(x) &= -6\Upsilon^4(x) + 12\Upsilon^3(x) - 7\Upsilon^2(x) + \Upsilon(x); \\ \varphi_4(x) &= 24\Upsilon^5(x) - 60\Upsilon^4(x) + 50\Upsilon^3(x) - 15\Upsilon^2(x) + \Upsilon(x); \\ \varphi_5(x) &= -120\Upsilon^6(x) + 360\Upsilon^5(x) - 390\Upsilon^4(x) + 180\Upsilon^3(x) - 31\Upsilon^2(x) + \Upsilon(x).\end{aligned}$$

O conjunto das funções PPS-wavelet é composto por funções pertencentes ao  $L^2(\mathbf{R})$ , que satisfazem a condição de admissibilidade de frame no espaço de Hilbert [3, 9].

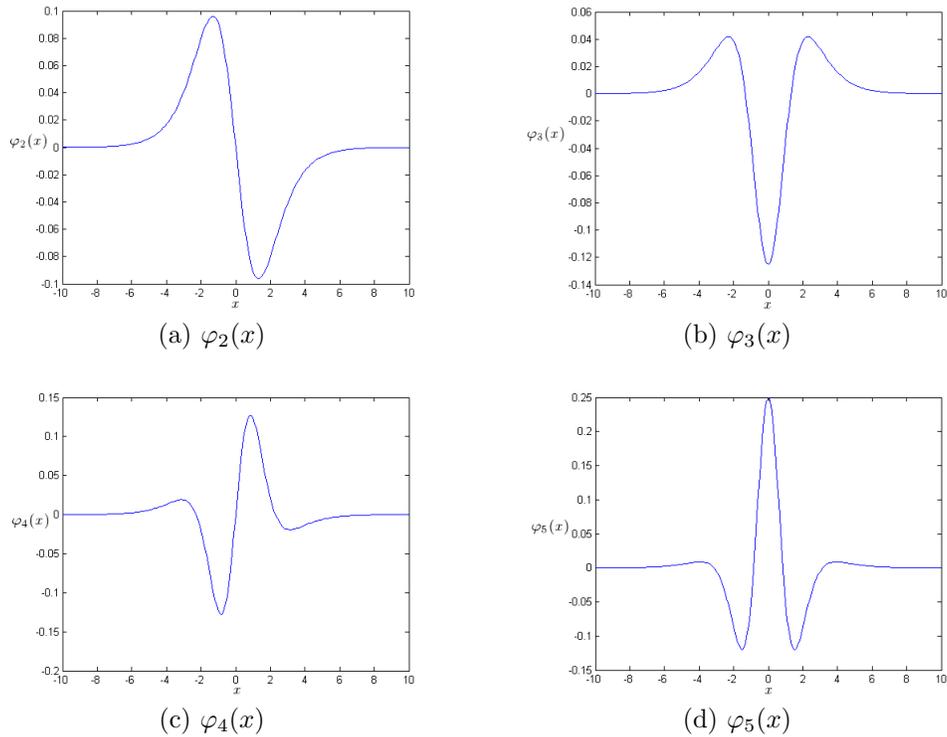


Figura 1: Ilustração das funções PPS-wavelet  $\varphi_2(x)$ ,  $\varphi_3(x)$ ,  $\varphi_4(x)$  e  $\varphi_5(x)$ .

### 3. Redes Neurais PPS-wavelet

Seja  $X_q = (x_{q1}, x_{q2}, \dots, x_{qm})$  uma amostra  $q$  de entrada,  $\varphi(x)$  uma função PPS-wavelet e  $\Upsilon(x)$  a função sigmóide definida na Eq. (2.1). A saída do  $j$ -ésimo neurônio da camada escondida é definida por

$$\varphi_{d_j, t_j} \left( \sum_{k=1}^m x_{qk} \right) = \varphi_{d_j, t_j} \left( net_{qj}^{(1)} \right) = \varphi \left( d_j \left( net_{qj}^{(1)} - t_j \right) \right),$$

onde

$$net_{qj}^{(1)} = \sum_{k=1}^m x_{qk}, \quad j = 1, 2, \dots, p.$$

O neurônio de saída é definido pela combinação linear dos  $p$  neurônios da camada escondida [9], isto é

$$f_q = \sum_{j=1}^p w_j \varphi_{d_j, t_j} \left( net_{qj}^{(1)} \right).$$

Sendo a saída da RNA PPS-wavelet  $F_q = (f_q)$  e a saída desejada  $Y_q = (y_q)$  para a amostra  $q$ , o erro total a ser minimizado durante o processo de treinamento da RNA PPS-wavelet para  $Q$  amostras é definido por

$$E = \frac{1}{2} \sum_{q=1}^Q (y_q - f_q)^2.$$

### 4. Inicialização da RNA PPS-wavelet

O problema de inicialização da RNA PPS-wavelet consiste em selecionar funções PPS-wavelet com as suas dilatações e translações que “melhor” mapeiam os dados de entrada, isto pode ser visto então como um problema de análise de regressão [2]. Neste sentido, são apresentados nesta seção três técnicas estudadas e implementadas para inicialização de arquiteturas neurais PPS-wavelets.

Seja  $\varphi(x)$  uma função PPS-wavelet, as técnicas utilizadas selecionam as funções PPS-wavelet  $\varphi_i^*(x)$  do conjunto

$$P = \left\{ \varphi_i^* : \varphi_i^*(x) = n_i \varphi(d_i(x - t_i)), \quad n_i = \left( \sum_{k=1}^N [\varphi(d_i(x_k - t_i))]^2 \right)^{-\frac{1}{2}}, \quad i = 1, \dots, L \right\}$$

onde  $x_1, \dots, x_N$  são as amostras de entrada,  $L$  é o número de funções PPS-wavelet,  $n_i$  está associado com a normalização da função PPS-wavelet e  $d_i, t_i$  correspondem aos parâmetros dilatação e translação da função PPS-wavelet  $\varphi(x)$ . As funções PPS-wavelet selecionadas são utilizadas na camada escondida da RNA PPS-wavelet.

A seleção de  $M \leq L$  funções PPS-wavelet do conjunto  $P$  baseado nos dados de treinamento de uma rede neural é um problema de análise de regressão definido por

[2, 15]

$$f_M(x) = \sum_{i=1}^M u_i \varphi_i^*(x),$$

onde os coeficientes reais  $u_i$  são obtidos com a minimização de

$$J = \frac{1}{N} \sum_{k=1}^N \left( y_k - \sum_{i=1}^M u_i \varphi_i^*(x_k) \right)^2. \quad (4.1)$$

Nos algoritmos apresentados a seguir, são utilizadas as seguintes notações matriciais

$$v_i = \begin{bmatrix} \varphi_i^*(x_1) \\ \vdots \\ \varphi_i^*(x_n) \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}, \quad (4.2)$$

onde  $v_i^T v_i = 1$  ( $i = 1, \dots, L$ ) e  $y_1, \dots, y_N$  são as saídas desejadas da RNA PPS-wavelet durante o treinamento. Seja  $i$  uma iteração dos algoritmos, denota-se por  $l_i$  o índice da função PPS-wavelet selecionada do conjunto  $P$ . Sendo  $i$  a iteração atual, então  $\varphi_{l_1}, \varphi_{l_2}, \dots, \varphi_{l_{i-1}}$  são funções PPS-wavelet selecionadas anteriormente.

#### 4.1. Seleção de funções PPS-wavelet baseada em resíduo

Esta técnica seleciona na primeira iteração a função PPS-wavelet  $\varphi_i^*(x)$  do conjunto  $P$  que “melhor” aproxima os dados de treinamento da rede neural, ou seja, a Eq. (4.1) assume o menor valor possível. Depois, iterativamente, seleciona a função PPS-wavelet que melhor aproxima o resíduo da aproximação do estágio anterior [15].

**Algoritmo: Seleção de funções PPS-wavelet baseada em resíduo**

Vetor resíduo inicial  $\gamma_0 \leftarrow {}^4y$ , com  $y$  definido na Eq. (4.2);  $f_0(x) \leftarrow 0$ ;

Para  $i = 1$  até  $M$  faça

início

seja  $I_i \leftarrow \{j : j = 1, \dots, L \text{ e } j \neq l_1, \dots, l_{i-1}\}$ ;

encontre  $l_i \leftarrow \arg \max_{j \in I_i} (v_j^T \gamma_{i-1})^2$  e determine

$$\begin{aligned} u_{l_i} &\leftarrow v_{l_i}^T \gamma_{i-1}; \\ f_i(x) &\leftarrow f_{i-1}(x) + u_{l_i} \varphi_{l_i}^*(x); \\ \gamma_i &\leftarrow \gamma_{i-1} - u_{l_i} v_{l_i}; \end{aligned}$$

fim

#### 4.2. Seleção de funções PPS-wavelet passo-a-passo por ortogonalização

Esta técnica considera a ortogonalidade das funções PPS-wavelet do conjunto  $P$ . Ela seleciona na primeira iteração a função PPS-wavelet que “melhor” aproxima os

<sup>4</sup>O símbolo  $\leftarrow$  indica um comando de atribuição.

dados de treinamento, então continua o processo de seleção levando em conta as funções PPS-wavelet já escolhidas [15].

Assumindo que no estágio  $i - 1$  já foram selecionadas as funções PPS-wavelet correspondentes aos vetores  $v_{l_1}, \dots, v_{l_{i-1}}$ . Para obtenção da  $i$ -ésima função PPS-wavelet, a técnica calcula a distância entre  $y$  e o espaço  $\Delta = \{v_{l_1}, \dots, v_{l_{i-1}}, v_j\}$  para cada  $j = 1, \dots, L$  e  $j \neq l_1, \dots, l_{i-1}$ . Para aumentar a eficiência computacional desta técnica, a ortogonalização dos vetores selecionados  $v_j$  é feita no final. Assume-se que  $v_{l_1}, \dots, v_{l_{i-1}}$  já são ortonormalizados e são renomeados como  $\omega_{l_1}, \dots, \omega_{l_{i-1}}$ .

Após  $M$  iterações do algoritmo, os valores de  $l_1, \dots, l_M$  são determinados, bem como  $\tilde{u}_{l_1}, \dots, \tilde{u}_{l_M}$  [15]. Porém, precisa-se ainda determinar os valores  $u_{l_1}, \dots, u_{l_M}$  em

$$f_M(x) = \sum_{i=1}^M u_i \varphi_{l_i}^*(x). \quad (4.3)$$

Estes valores  $u_i$  podem ser obtidos resolvendo o sistema de equações [15]

$$A[u_{l_1}, \dots, u_{l_M}]^T = [\tilde{u}_{l_1}, \dots, \tilde{u}_{l_M}]^T, \quad (4.4)$$

onde  $A$  é uma matriz triangular superior definida por

$$A = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \cdots & \cdots & \alpha_{1M} \\ 0 & \alpha_{22} & \alpha_{23} & \cdots & \cdots & \alpha_{2M} \\ 0 & 0 & \alpha_{33} & \cdots & \cdots & \alpha_{3M} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \alpha_{M-1M-1} & \alpha_{M-1M} \\ 0 & 0 & \cdots & 0 & 0 & \alpha_{MM} \end{bmatrix},$$

onde

$$\begin{aligned} \alpha_{ki} &\leftarrow v_{l_i}^T \omega_{l_k}, \quad (k = 1, \dots, i-1); & \alpha_{ii} &\leftarrow \left( \left( p_{l_i}^{(i)} \right)^T p_{l_i}^{(i)} \right)^{\frac{1}{2}}; \\ p_j &\leftarrow v_j - \left( (v_j^T \omega_{l_1}) \omega_{l_1} + \cdots + (v_j^T \omega_{l_{i-1}}) \omega_{l_{i-1}} \right). \end{aligned}$$

**Algoritmo: Seleção de funções PPS-wavelet passo-a-passo por ortogonalização**

Seja  $I_1 = \{1, 2, \dots, L\}$ , encontre  $l_1 \leftarrow \arg \max_{j \in I_1} (v_j^T y)^2$  e calcule

$$\tilde{u}_{l_1} \leftarrow v_{l_1}^T y; \quad \omega_{l_1} \leftarrow v_{l_1};$$

$$\alpha_{11} \leftarrow 1; \quad p_j^{(1)} \leftarrow v_j, \quad j = 1, \dots, L, \quad j \neq l_1;$$

Para  $i \leftarrow 2$  até  $M$  faça  
início

Seja  $I_i = I_{i-1} - \{l_{i-1}\}$ . Para cada  $j \in I_i$ , calcule

$$\begin{aligned} p_j^{(i)} &\leftarrow p_j^{(i-1)} - (v_j^T \omega_{l_{i-1}}) \omega_{l_{i-1}}; \\ I_i &\leftarrow I_i - \{j : p_j^{(i)} = 0\}; \end{aligned}$$

Encontre  $l_i \leftarrow \arg \max_{j \in I_i} \frac{\left( (p_j^{(i)})^T y \right)^2}{\left( (p_j^{(i)})^T p_j^{(i)} \right)}$  e determine

$$w_{l_i} \leftarrow \left( (p_{l_i}^{(i)})^T p_{l_i}^{(i)} \right)^{-\frac{1}{2}} p_{l_i}^{(i)}; \quad \tilde{u}_{l_i} \leftarrow w_{l_i}^T y;$$

$$\alpha_{ki} \leftarrow v_{l_i}^T \omega_{l_k}, \quad (k = 1, \dots, i-1); \quad \alpha_{ii} \leftarrow \left( (p_{l_i}^{(i)})^T p_{l_i}^{(i)} \right)^{\frac{1}{2}};$$

fim

Resolva a Eq. (4.4) para  $u_{l_i}$ ,  $i = 1, \dots, M$ ;

Calcule  $f_M(x)$  definido na Eq. (4.3).

### 4.3. Eliminação recorrente de funções PPS-wavelet

Inicialmente, esta técnica seleciona todas as funções PPS-wavelet do conjunto  $P$  conforme representado na Eq. (4.5), então funções PPS-wavelet são eliminadas em cada iteração do algoritmo, visando aumentar o resíduo [15].

$$f_L(x) = \sum_{i=1}^L u_i \varphi_i^*(x). \quad (4.5)$$

Os resíduos são definidos por  $\gamma_L(k) = y_k - f_L(x_k)$ ,  $k = 1, \dots, N$ ; e podem ser reescritos segundo a forma vetorial  $\gamma_L = y - (v_1, \dots, v_L) (u_1, \dots, u_L)^T$ .

Seja a matriz  $V_0 = (v_1, \dots, v_L)$ , determina-se  $\gamma_L^T \gamma_L = y^T y^T - y^T V_0 (V_0^T V_0)^{-1} V_0^T y$ . Com a remoção de uma função PPS-wavelet, diz-se  $\varphi_j^*$  de  $f_L(x)$  tem-se

$$\gamma_{L-1}^T \gamma_{L-1} = y^T y^T - y^T C(v_j|V_0) (C(v_j|V_0)^T C(v_j|V_0))^{-1} C(v_j|V_0)^T y,$$

onde o operador  $C$  é o complemento da matriz, por exemplo, se a matriz  $U = [U_1, U_2, U_3]$ , então  $C(U_2|U) = [U_1, U_3]$ . O incremento do resíduo causado pela remoção de  $\varphi_j^*$  de  $f_L(x)$  é definido por

$$J(\varphi_j^*) = y^T V_0 (V_0^T V_0)^{-1} V_0^T y - y^T C(v_j|V_0) \cdot (C(v_j|V_0)^T C(v_j|V_0))^{-1} C(v_j|V_0)^T y \quad (4.6)$$

A remoção da função PPS-wavelet  $\varphi_j^*$  de  $f_L(x)$ , que minimiza a Eq. (4.6), resulta em  $f_{L-1}(x)$ . Repete-se a mesma rotina para remover uma outra função PPS-wavelet de  $f_{L-1}(x)$ , e assim por diante.

#### Algoritmo: Eliminação recorrente de funções PPS-wavelet

Seja o conjunto com as funções PPS-wavelet:  $V_0 \leftarrow (v_1, \dots, v_L)$ ;

Para  $i \leftarrow 1, \dots, L - M$  faça

início

Seja  $I_i \leftarrow \{j : j = 1, \dots, L \text{ e } j \neq l_1, \dots, l_{i-1}\}$ ;  
 Calcule  $u \leftarrow (V_{i-1}^T V_{i-1})^{-1} V_{i-1}^T y$ , onde  $u$  é um vetor composto de  $u_j, j \in I_i$ ;  
 Encontre  $l_i \leftarrow \arg \min_{j \in I_i} u_j^2$ ;  
 Determine  $V_i \leftarrow C(v_j | V_{i-1})$ ;  
 fim  
 Seja  $I_{L-M+1} \leftarrow \{j : j = 1, \dots, L \text{ e } j \neq l_1, \dots, l_{L-M}\}$ ;  
 Determine  $f_M(x) \leftarrow \sum_{j \in I_{L-M+1}} u_j \varphi_j^*(x)$  com os componentes  $u_j$  do vetor  $u$  definido por

$$u \leftarrow (V_{L-M}^T V_{L-M})^{-1} V_{L-M}^T y.$$

## 5. Conclusões

Em redes neurais artificiais, em especial, nas redes neurais PPS-wavelets é ainda muito difícil definir a arquitetura ideal para um dado problema; de maneira que ela seja tão grande quanto o necessário para conseguir obter a representação adequada do problema e, ao mesmo tempo, pequena o suficiente para realizar um treinamento mais rápido. Dentro deste cenário, foram investigadas algumas técnicas metodológicas para a seleção de neurônios em redes neurais PPS-wavelets com o objetivo claro de estabelecer melhorias junto ao processo de inicialização. Resultados experimentais vem demonstrando a melhoria da sua convergência através desta metodologia. Os arquivos fontes para o uso destas implementações encontram-se em <http://saci.fc.unesp.br/rafael/algoritmos.zip> para testes, críticas e sugestões. Assim, este trabalho constitui-se em mais uma nova alternativa para o desenvolvimento da área de redes neurais artificiais.

### Agradecimentos

O primeiro autor, mestrando no ICMC-USP, agradece o apoio financeiro parcial do PIBIC-CNPq, Processo 108250/2005-9. Também, agradece à FAPESP pela bolsa de mestrado, Processo 2006/05910-1. O segundo autor agradece o suporte financeiro da CAPES, Processo 3634/06-0.

### Referências

- [1] W. Bellil, C.B. Amar, A.M. Alimi, Comparison between beta wavelets neural networks, RBF neural networks and polynomial approximation for 1D, 2D functions approximation, *Trans. on Eng., Comp. and Tech.*, **13** (2006), 102-107.
- [2] S. Chen, S. Billings, W. Luo, Orthogonal least squares learning methods and their application to non-linear system identification, *Ind. J. Control*, **50** (1989), 1873-1896.
- [3] C.S. Chui, "An Introduction to Wavelets", Academic Press, 1992.

- [4] G. Cybenko, Approximation by superpositions of a sigmoidal function, *Math. of Control, Signals, and Systems*, **2** (1989), 303-314.
- [5] S. Haykin, "Redes Neurais: Princípios e Práticas", Bookman, 2001.
- [6] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, *Neural Networks*, **2** (1989), 359-366.
- [7] K.Z. Mao, H. Guang-Bin, Neuron selection for RBF neural network classifier based on data structure preserving criterion, *IEEE Trans. on Neural Networks*, **16** (2005), 1531-1540.
- [8] J.F. Marar, E.C.F. Carvalho, G.C. Vasconcelos, Function approximation by polynomial wavelets generated from powers of sigmoids, *SPIE - Wavelet Application III*, **2762** (1996), 365-374.
- [9] J.F. Marar, "Polinômios Potências de Sigmóide (PPS): Uma Nova Técnica para Aproximação de Funções, Construção de Wavenets e Suas Aplicações em Processamento de Imagens e Sinais", Tese de Doutorado, UFPE, Recife, PE, 1997.
- [10] J.F. Marar, E.C.B. Carvalho Filho, J.D. Santos, Mathematical tests about the existence and applications of PPS wavelets in function approximation problems, in *SPIE*, **3391** (1998), 455-466.
- [11] X. Mei, S.-H. Sun, Comparison of neuron selection algorithms of wavelet-based neural network, *SPIE - Neural Network and Distributed Processing*, **4555** (2001), 121-126.
- [12] R.A.B. Queiroz, J.F. Marar, Algoritmo adaptativo baseado no método Levenberg-Marquardt para treinamento de Redes Neurais PPS-wavelet: Aproximação de funções-um estudo de caso, em "XI Encontro de Iniciação Científica e Pós-Graduação do Instituto Tecnológico de Aeronáutica", 2005.
- [13] Q. Zhang, A. Benveniste, Wavelet Networks, *IEEE Trans. on Neural Networks*, **3** (1992).
- [14] Q. Zhang, Regressor selection and wavelet network construction, *32nd IEEE Conference on Decision and Control*, **4** (1993), 3688-3693.
- [15] Q. Zhang, Using wavelet network in nonparametric estimation, *IEEE Trans. on Neural Networks*, **8** (1997), 227-236.