

Resolução de um modelo de reator de leito fixo não adiabático com dispersão axial utilizando redes neurais artificiais

Luiz Henry Monken e Silva, Ivo Neitzel e Ed Pinheiro Lima*

Programa de Pós-Graduação em Engenharia Química, Universidade Estadual de Maringá, Maringá, Paraná, Brasil. *Autor para correspondência. e-mail: edpl@deq.uem.br

RESUMO. As capacidades de interpolação de redes perceptron multicamada (MLP) foram utilizadas para resolver um sistema de equações diferenciais ordinárias que modela um reator não-adiabático com leito fixo e dispersão axial. As metodologias descritas neste artigo seguem as propostas por Lagaris *et al.* (1998, 2000), estendidas para modelos com condições de contorno mistas e pelo uso do método da penalidade para converter o problema de otimização original de restrito para irrestrito no treinamento das redes MLP. Os resultados são compatíveis com aqueles apresentados em Luize e Biscaia (1991), que foram obtidos com técnicas numéricas já consagradas, como elementos finitos e colocação ortogonal. O método de neuro-interpolação adotado neste artigo é de fácil manuseio se comparado com os métodos clássicos para solução numérica de equações diferenciais, particularmente para sistemas diferenciais não-lineares, e define uma aproximação global, na forma analítica, para a solução de problemas.

Palavras chave: redes neurais, equações diferenciais.

ABSTRACT. Model resolution of an axial dispersed non-adiabatic fixed bed reactor using artificial neural networks. The interpolation capabilities of multilayer perceptron networks (MLP) were used to solve a system of ordinary differential equations that models an axial dispersed non-adiabatic fixed bed reactor. The methodologies described in this paper follow the first ones proposed by Lagaris *et al.* (1998, 2000), but enlarge them to differential models with mix boundary conditions and by the use of the penalty method to convert the original constrained to unconstrained optimization problem in training the MLP networks. The results are in agreement on those in Luize e Biscaia (1991), which were obtained by well-established numerical techniques as finite element and orthogonal collocation methods. The neural interpolation method used in this paper is easier to handle than the classical methods for numerical solution of differential equations, particularly for non-linear differential systems, and defines a global approximation, in analytic form, for problems solution.

Key words: neural networks, differential equations.

Introdução

As capacidades de aproximação de funções por meio de redes neurais artificiais são bem conhecidas desde o final da década de 1980, como se pode constatar pelos resultados alcançados por Cybenko (1989), Funahashi (1989), Hornik *et al.* (1989), Girosi e Poggio (1990) e Cotter (1990); todos precedidos pelos resultados de Komolgorov (1957, *apud* Funahashi, 1989) e de Sprecher (1965, *apud* Funahashi, 1989). Em síntese, qualquer função mensurável pode ser aproximada por uma rede neural progressiva de camada escondida simples.

Apesar desses resultados teóricos e de suas variadas aplicações, são recentes os primeiros trabalhos que usam tais capacidades para resolução de equações ou sistemas de equações diferenciais

ordinárias – problema de valor inicial e de valor no contorno – e de equações diferenciais parciais – problemas estacionários, a valor no contorno, como se pode ver em Lagaris *et al.* (1998, 2000) e Wang e Lin (1998).

Este trabalho tem por objetivo ampliar o uso dessa metodologia de interpolação para resolução de sistemas diferenciais que aparecem em problemas de relevância técnica em engenharia química. Em vista disto, foi escolhido, pela suas características matemáticas, numérica e computacional, um modelo de reator de leito fixo não-adiabático com dispersão axial para ser simulado numericamente por redes neurais. Convém citar que uma alternativa para resolução analítica desse problema é a solução por séries, que se mostra de convergência lenta.

Na segunda seção do trabalho, é apresentado o modelo matemático do reator a ser resolvido, com a descrição sucinta das variáveis e dos parâmetros intervenientes. A metodologia de interpolação por redes neurais e de resolução numérica do modelo consta da terceira seção. Na seção seguinte, são expostos os experimentos numéricos e os resultados que podem ser vistos em Luize e Biscaia (1991), obtidos por métodos já bem mais estabelecidos, como o de elementos finitos e o de colocação ortogonal. As conclusões, na última seção, referem-se ao método propriamente dito, quanto a sua facilidade de uso, aplicabilidade e eficiência computacional.

Material e métodos

Modelo matemático de reator de leito fixo não-adiabático

Para efeito de usar interpolação por redes neurais para resolução de equações diferenciais ordinárias, foi escolhido um modelo matemático de reator de leito fixo não adiabático, descrito por Luize (1991), o qual, para facilitar o entendimento dos procedimentos numéricos, é aqui transcrito:

$$\frac{1}{Pe_m} \frac{d^2 C}{dx^2} - \frac{dC}{dx} - R(C, T) = 0, \quad 0 < x < 1 \quad (1)$$

$$\frac{1}{Pe_h} \frac{d^2 T}{dx^2} - \frac{dT}{dx} + \beta R(C, T) + H_w (T_w - T) = 0, \quad (2)$$

sendo

$$R(C, T) = Da e^{\gamma \left(1 - \frac{1}{T}\right)} C(x)^m, \quad (3)$$

em que

x	comprimento adimensional do reator
C	concentração adimensional do reagente
T	temperatura adimensional do reagente
Pe_m	número de Peclet de massa
Pe_h	número de Peclet de energia
Da	número de Damköhler
β	adimensional do calor de reação
H_w	adimensional do coeficiente de transferência de calor
T_w	temperatura adimensional na parede do reator
γ	adimensional da energia de ativação
m	ordem da reação química

sujeito às condições de contorno,

$$\left. \frac{1}{Pe_m} \frac{dC}{dx} \right|_{x=0} - C(x) \Big|_{x=0} = -1, \quad (4)$$

$$\left. \frac{1}{Pe_h} \frac{dT}{dx} \right|_{x=0} - T(x) \Big|_{x=0} = -1, \quad (5)$$

$$\left. \frac{1}{Pe_m} \frac{dC}{dx} \right|_{x=1} = 0, \quad (6)$$

$$\left. \frac{1}{Pe_h} \frac{dT}{dx} \right|_{x=1} = 0. \quad (7)$$

Portanto, o modelo matemático é formado por um sistema de equações diferenciais ordinárias de segunda ordem não-lineares, (1) e (2), e pelas condições de contorno, (4)-(7). Ele foi simulado usando métodos clássicos, como o de elementos finitos, diferenças finitas e o de colocação, e os resultados obtidos podem ser vistos em Luize (1991). O mesmo modelo é aqui resolvido, por redes neurais, para quatro conjuntos de parâmetros. Nas próximas seções, os resultados alcançados pela interpolação por redes neurais serão apresentados e comparados com os obtidos por elementos finitos e pela colocação ortogonal.

Descrição do método e implementação numérica

O sistema diferencial (1)-(2) insere-se na seguinte forma diferencial implícita geral:

$$G_1(x, C'', C', C, T) = F_1(x) \quad (8)$$

$$G_2(x, T'', T', T, C) = F_2(x), \quad (9)$$

cujas equações são definidas, sem perda de generalidade, no intervalo $0 < x < 1$.

Para a resolução numérica desse ou de outros sistemas com mais equações, é adotado um procedimento de colocação, similar ao procedimento clássico, que pode ser visto em Douglas e Dupont (1973) e Lapidus e Pinder (1982).

Inicialmente, a formulação diferencial (8)-(9) do problema é levada a uma formulação fraca, ver Dautray e Lions (1984), resultando

$$\int_0^1 G_1(x, C'', C', T) v(x) dx = 0, \quad (10)$$

e

$$\int_0^1 [G_2(x, T'', T', T, C) - F(x)] w(x) dx = 0, \quad (11)$$

em que v e w são funções testes.

O método da colocação segue com a escolha no domínio $0 < x < 1$ de N pontos distintos x_i , $1 \leq i \leq N$, nos quais todos os termos presentes em (8) e (9) estão definidos. Em seguida, as variáveis dependentes C e T são aproximadas por C_H e T_H e em um espaço de dimensão finita, que, no caso, depende da arquitetura da rede neural a ser usada. Por sua vez, o espaço das funções testes é um espaço de N distribuições de Dirac $\psi_i(x) = \delta(x - x_i)$, $1 \leq i \leq N$. Tomando v e w iguais, para usar os mesmos pontos de colocação para o sistema, e como sendo essas distribuições, as equações (10) e (11), na forma discreta, ficam sendo,

$$\int_0^1 G_1(x, C_H'', C_H', T_H) \psi_i(x) dx = 0, \quad (12)$$

$$\int_0^1 [G_2(x, T_H'', T_H', T_H, C_H) - F(x)] \psi_i(x) dx = 0, \quad (13)$$

que pela definição da função ψ_i fornecem os resíduos nos pontos de colocação:

$$R_1(x_i, \bar{p}_1) = G_1[x_i, C_H''(x_i), C_H'(x_i), T_H(x_i)] = 0 \quad (14)$$

e

$$R_2(x_i, \bar{p}_2) = G_2[x_i, T_H''(x_i), T_H'(x_i), T_H(x_i), C_H(x_i)] - F(x_i) = 0 \quad (15)$$

sendo $1 \leq i \leq N$ e $\bar{p}_1$ e $\bar{p}_2$ vetores formados pelos pesos e *bias* das redes neurais (daqui para frente referidos genericamente como pesos). Nas equações (12)-(15), as funções C_H e T_H usam aproximações por redes neurais que adiante serão definidas.

A solução do sistema não-linear formado pelas equações (12)-(15) é levada ao contexto de treinamento de redes neurais. Primeiro é definido um funcional a partir da soma dos quadrados dos resíduos, tendo-se nesse modelo,

$$E(\bar{p}) = \sum_{i=1}^N \{[R_1(x_i, \bar{p}_1)]^2 + [R_2(x_i, \bar{p}_2)]^2\}, \quad (16)$$

em que, $\bar{p} = [\bar{p}_1, \bar{p}_2]^T$. Os pesos serão determinados resolvendo-se o problema de otimização sob restrições impostas pelas condições

de contorno (4)-(7). Isso é feito com o auxílio do método da penalidade, pela ponderação das restrições e redefinição do funcional (16). Assim,

$$E(\bar{p}) = \sum_{i=1}^N \{[R_1(x_i, \bar{p}_1)]^2 + [R_2(x_i, \bar{p}_2)]^2\} + \eta \{[RB_1(0)]^2 + [RB_2(0)]^2 + [RB_3(1)]^2 + [RB_4(1)]^2\} \quad (17)$$

em que, η é o parâmetro de penalidade e os resíduos no contorno são:

$$RB_1(0) = \frac{1}{Pe_m} \frac{dC_H}{dx} \Big|_{x=0} - C_H(x) \Big|_{x=0} - 1 = 0, \quad (18)$$

$$RB_2(0) = \frac{1}{Pe_h} \frac{dT_H}{dx} \Big|_{x=0} - T_H(x) \Big|_{x=0} - 1 = 0, \quad (19)$$

$$RB_3(1) = \frac{1}{Pe_m} \frac{dC_H}{dx} \Big|_{x=1} = 0, \quad (20)$$

$$RB_4(1) = \frac{1}{Pe_h} \frac{dT_H}{dx} \Big|_{x=1} = 0. \quad (21)$$

Dessa forma, o problema de otimização passa a ser sem restrições, isto é,

$$\min_{\bar{p}} E(\bar{p}) = \sum_{i=1}^N \{[R_1(x_i, \bar{p}_1)]^2 + [R_2(x_i, \bar{p}_2)]^2\} + \eta \{[RB_1(0)]^2 + [RB_2(0)]^2 + [RB_3(1)]^2 + [RB_4(1)]^2\} \quad (22)$$

e sua solução foi procurada com métodos clássicos de otimização, mediante implementação numérica apropriada como descrito a seguir.

Implementação numérica

A minimização do funcional erro definido em (22) foi realizada com a concatenação das rotinas DUMCGF e DUMINF, disponíveis no *IMSL Numerical Libraries*. Embora o método usado na DUMINF seja de convergência quadrática, ela não se mostrou robusta o suficiente para partir de pesos arbitrários, quando o número de parâmetros de otimização foi superior a cerca de 40, ou seja, uma rede progressiva 2:10:1 (2 entradas, 10 neurônios na camada escondida, e 1 saída), arquitetura muito usada para solução de equações diferenciais e em particular no modelo em consideração.

No procedimento de aproximação das variáveis dependentes, são utilizadas as seguintes redes perceptron multicamada:

$$C_H(x, \bar{p}_1) = \sum_{k=1}^H v_k \sigma(w_k x + b_k), \quad (23)$$

$$T_H(x, \bar{p}_2) = \sum_{k=1}^H r_k \sigma(s_k x + c_k), \quad (24)$$

em que

$$\bar{p}_1 = [v_1, w_1, b_1, \dots, v_H, w_H, b_H]^T \text{ e } \bar{p}_2 = [r_1, s_1, c_1, \dots, r_H, s_H, c_H]^T$$

Tendo sido escolhido o número de unidades escondidas $H=10$, e como função de ativação dos neurônios da camada escondida a função sigmoideal definida por,

$$\sigma(x) = \frac{1}{1 + \exp(-x)}, \quad (25)$$

Com as redes (23) e (24), a dimensão do espaço de interpolação é de 60 parâmetros. Nas simulações numéricas realizadas foram utilizados $N=40$ pontos de colocação, $x_i, 1 \leq i \leq N$, igualmente espaçados, no intervalo $[0,1]$.

Para aproveitar as melhores características computacionais das sub-rotinas DUMCGF e DUMINF, o processo de otimização é feito em duas etapas: inicialmente, com pesos arbitrários a otimização é feita pela DUMCGF, executando cerca de 7500 iterações, que demanda pouco tempo (em computadores com processadores de 800 MHz e 384 MB memória). Em seguida, os pesos finais da primeira etapa servem como pesos iniciais para a etapa de refinamento da otimização pela DUMINF.

Resultados e discussão

O modelo matemático, equações (1)-(7), foi resolvido para quatro conjuntos de parâmetros, seguindo as simulações realizadas por Luize (1991). Os resultados para a concentração do reagente estão mostrados nas Tabelas 1 a 4, sendo que cada caso corresponde a um conjunto de parâmetros que se encontram especificados nas próprias tabelas. Nelas também são reproduzidos os resultados de Luize e Biscaia (1991), obtidos por elementos finitos e colocação ortogonal. Os perfis de distribuição da concentração do reagente e da temperatura ao longo do reator estão mostrados nas Figuras 1 a 4.

Tabela 1. Concentração no início e no final do reator no caso 1.

$Pc_m=1,0$ $Pc_h=1,0$ $m=1,0$ $Da=0,1$ $\beta=0,0010$ $\gamma=1,0$ $H_w=0,0$ $T_w=0,0$					
Ponto	Elementos finitos ⁽¹⁾	Colocação ortogonal ⁽²⁾	Rede Neural	Diferença 1	Diferença 2
0,0	0,94165	0,94172	0,94165	0,0	7E-5
1,0	0,90799	0,90808	0,90799	0,0	9E-5

Fonte: Luize e Biscaia, (1991).

(1) 20 pontos de colocação (2) 40 pontos de colocação

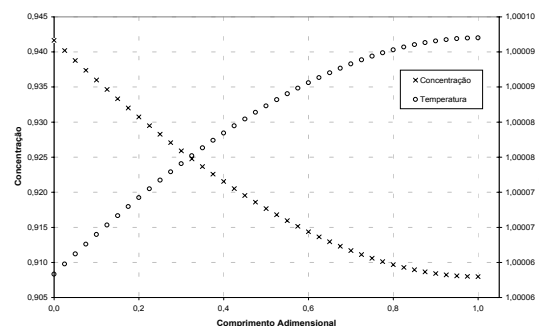


Figura 1. Gráfico da concentração e da temperatura do caso 1.

Tabela 2. Concentração no início e no final do reator no caso 2.

$Pc_m=2,0$ $Pc_h=2,0$ $m=1,0$ $Da=0,5$ $\beta=0,0100$ $\gamma=2,0$ $H_w=0,0$ $T_w=0,0$					
Ponto	Elementos finitos ⁽¹⁾	Colocação ortogonal ⁽²⁾	Rede Neural	Diferença 1	Diferença 2
0,0	0,83767	0,83880	0,83769	2E-5	1,11E-3
1,0	0,64110	0,64321	0,64107	3E-5	2,14E-3

Fonte: Luize (1991).

(1) 20 pontos de colocação (2) 40 pontos de colocação

Em todos os casos, foram dados dois passos: um número de iterações, geralmente 7.500, para a rotina DUMCGF e o refino dos pesos resultantes com a rotina DUMINF. Um resumo das execuções para cada caso está mostrado na Tabela 5 e pode ser encontrada em Lima et al. (2002).

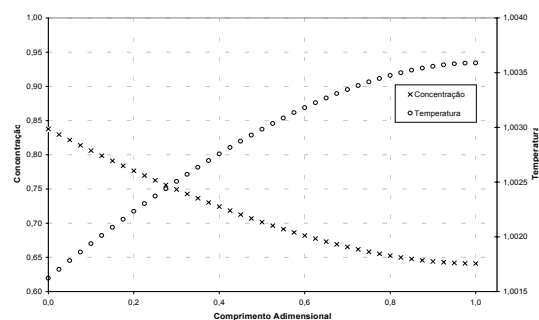


Figura 2. Gráfico da concentração e da temperatura do caso 2.

Tabela 3. Concentração no início e no final do reator no caso 3.

$Pc_m=5,0$ $Pc_h=5,0$ $m=1,0$ $Da=1,0$ $\beta=0,0500$ $\gamma=10,0$ $H_w=0,0$ $T_w=0,0$					
Ponto	Elementos finitos ⁽¹⁾	Colocação ortogonal ⁽²⁾	Rede Neural	Diferença 1	Diferença 2
0,0	0,83950	0,84145	0,83949	1,00e-5	1,96e-3
1,0	0,34267	0,35020	0,34257	1,00e-4	7,63E-3

Fonte: Luize (1991).

(1) 20 pontos de colocação (2) 40 pontos de colocação

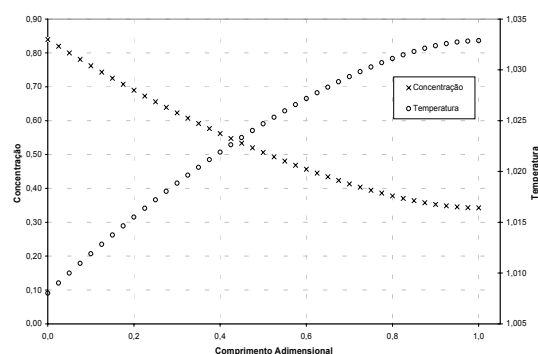


Figura 3. Gráfico da concentração e da temperatura do caso 3.

Tabela 4. Concentração no início e no final do reator no caso 4.

$Pc_{in}=96,0$ $Pc_{f}=96,0$ $m=1,0$ $Da=3,8$ $\beta=0,0560$ $\gamma=17,6$ $H_w=1,0$ $T_w=1,0$					
Ponto	Elementos finitos ⁽¹⁾	Colocação ortogonal ⁽²⁾	Rede Neural	Diferença 1	Diferença 2
0,0	0,96024	0,95618	0,97564	1,54 E-2	1,95 E-2
1,0	0,00039	0,00054	0,00112	7,30 E-4	5,80 E-4

Fonte: Luize (1991)

⁽¹⁾ 20 pontos de colocação ⁽²⁾ 40 pontos de colocação

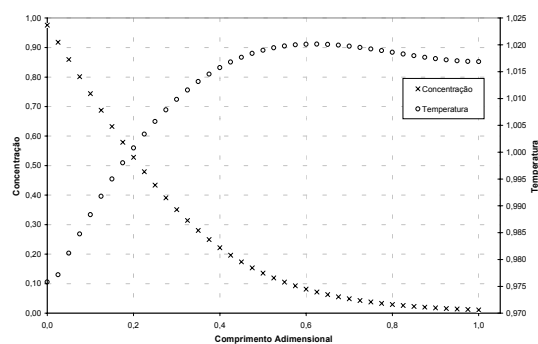


Figura 4. Gráfico da concentração e da temperatura do caso 4.

Tabela 5. Resumo das execuções para os quatro casos.

Caso	Passo 1 ⁽¹⁾		Passo 2 ⁽²⁾
	Iterações	Tempo (min)	Tempo (min)
1	7.500	20	15
2	7.500	22	12
3	7.500	19	27
4	65.000	318	9

⁽¹⁾ rotina DUMCGF ⁽²⁾ rotina DUMINF

Pela análise desses resultados, constata-se que, nos três primeiros casos, o método da rede neural utilizado apresenta a mesma precisão, em até duas casas decimais, em relação aos métodos de elementos finitos e de colocação ortogonal. No quarto caso, os parâmetros existentes no sistema de equações exigem mais dos três métodos. No caso dos métodos clássicos, a precisão foi reduzida para uma casa decimal o que também se verificou com o método utilizado.

Conclusão

Um modelo matemático para um reator de leito fixo adiabático com dispersão axial foi objeto de simulação numérica pelo emprego de metodologia de redes neurais. A utilização de rede neural, na solução de equações diferenciais, é recente (os primeiros trabalhos datam de 1998), daí a necessidade de se ampliar suas aplicações a problemas técnicos relevantes, em particular da engenharia química, para se ter acesso às suas características numéricas e computacionais.

As simulações realizadas seguiram as de Luize e Biscaia (1991) e os resultados obtidos pelo método proposto são tão precisos quanto os alcançados por elementos finitos e de colocação ortogonal, amplamente empregados em simulações numéricas de problemas de ciências e da engenharia. Em particular, as aplicações aqui tratadas ampliam o método proposto por Lagaris *et al.* (1998), pelo uso do método da penalidade de fácil adaptação para inserir restrições oriundas de condições de contorno ou de outros tipos de restrições, e pela resolução de sistema de equações ordinárias com condições de contorno do tipo misto, como as do modelo descrito em (1)-(7).

O que ainda se pode constatar do método de resolução de equações diferenciais por redes neurais artificiais é que o tipo de aproximação é global. Após o treinamento da rede neural, tem-se uma função de interpolação que descreve, para qualquer ponto do domínio, diretamente, uma aproximação para a solução da equação diferencial.

Por fim, uma significativa característica em relação à utilização de redes neurais para resolução de equações diferenciais é sobre a linearidade ou não do modelo. O método de redes neurais não necessita de modificação em sua metodologia numérica e de implementação computacional, ao resolver quer problemas lineares quer não lineares.

Referências

- COTTER, N.E. The Stone-Weierstrass theorem and its application to neural networks. *IEEE Trans. Neural Netw.*, New York, v. 1, n. 4, p. 290-295, 1990.
- CYBENKO, G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, Godalming, v. 2, n. 4, p. 303-314, 1989.
- DAUTRAY, R.; LIONS, J.L. *Analyse mathématique et calcul numérique*. Paris: Masson Editeur, v. 6, 1984.
- DOUGLAS JR, J.; DUPONT, T. A Finite element collocation method for quasilinear parabolic equations. *Mathematics of Computation*, Providence, v. 27, n. 121, p. 17-28, 1973.

- FUNAHASHI, K. On the Approximate realization of continuous mappings by neural networks neuro-control systems theory and applications. *Neural Netw.*, Exeter, v. 2, n. 3, p. 183-192, 1989.
- GIROSI, F.; POGGIO, T. Networks and the Best Approximation property. *Biol. Cybern.*, v. 63, p. 169-176, 1990.
- HORNIK, K. et al. Multilayer Feedforward Networks are Universal Approximators. *Neural Netw.*, Exeter, v. 2, n. 5, p. 359-366, 1989.
- LAGARIS, I.E. et al. Artificial neural network for solving ordinary and partial differential equations. *IEEE Trans. Neural Netw.*, New York, v. 9, n. 5, p. 987-1000, 1998.
- LAGARIS, I.E. et al. Neural-network methods for boundary value problems with irregular boundaries. *IEEE Trans. Neural Netw.*, New York, v. 11, n. 5, p. 1041-1049, 2000.
- LAPIDUS, L.; PINDER, G.F. *Numerical solution of partial differential equations in Science and Engineering*. New Jersey: John Wiley & Sons, 1982.
- LIMA, E.P. *Soluções de modelos diferenciais discretos ou contínuos por redes neurais artificiais: contornos irregulares*. 2002. Dissertação (Mestrado) – Universidade Estadual de Maringá, Maringá, 2002.
- LUIZE, A. *Avaliação de técnicas numéricas para a resolução de problemas de difusão com reação química*. 1991. Dissertação (Mestrado) – COPPE/Universidade Federal do Rio de Janeiro, Rio de Janeiro, 1991.
- WANG, Y.; LIN, C. Runge-Kutta neural network for identification of dynamical systems in high accuracy. *IEEE Trans. Neural Netw.*, New York, v. 9, n. 2, p. 266-280, 1998.
- Received on February 16, 2003.
Accepted on June 02, 2003.