

INCORPORAÇÃO DE RECORRÊNCIA EM ESTRUTURAS NEURAS NEBULOSAS

Ivette Luna¹, Rosangela Ballini², Fernando Gomide³

^{1,3} Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
13083-970 Campinas – SP – Brasil

E-mail: iluna@densis.fee.unicamp.br; gomide@dca.fee.unicamp.br

² Universidade Estadual de Campinas
Instituto de Economia
13083-970 Campinas – SP – Brasil
E-mail: ballini@eco.unicamp.br

Resumo – Neste artigo uma classe de estruturas de redes neurais nebulosas recorrentes é proposta. Os modelos neurais apresentados são compostos de duas partes: um sistema de inferência nebuloso e uma rede neural clássica. O sistema nebuloso é formado por neurônios lógicos modelados através de operadores *AND* e *OR*, utilizando normas triangulares. Estes neurônios compõem a camada intermediária. A rede neural é formada por neurônios clássicos com funções de ativação não lineares em série com as unidades lógicas prévias. O sistema de inferência nebuloso codifica um conjunto de regras do tipo se-então, sendo a inferência efetuada pelos neurônios lógicos onde ocorre a recorrência. Os pesos da camada intermediária são ajustados utilizando um algoritmo de treinamento por reforço associativo e os pesos da saída ajustados via o gradiente do erro quadrático. As redes neurais nebulosas recorrentes propostas constituem um meio efetivo para modelagem de sistemas não lineares. Resultados de simulação envolvendo a identificação de um sistema dinâmico não linear mostram que as estruturas propostas fornecem modelos simples, com um mecanismo de aprendizado rápido e erros de aproximação baixos quando comparados com modelos da literatura.

Palavras-Chave - Redes neurais, redes neurais nebulosas, redes recorrentes, modelagem de sistemas.

1. Introdução

Nestes últimos anos, sistemas neurais nebulosos tornaram-se uma alternativa atrativa para aplicações em áreas distintas da ciência e engenharia, devido a sua capacidade de tratar imprecisão e incerteza presentes na informação disponível e na dinâmica dos próprios sistemas.

Sistemas neurais nebulosos combinam dois grandes paradigmas: redes neurais artificiais e teoria de conjuntos nebulosos. Em particular, sistemas neurais nebulosos estáticos se mostram promissores em identificação de sistemas, controle, previsão de séries temporais, classificação de padrões, etc. Contudo, o uso de redes neurais nebulosas estáticas ainda é restrito, devido à sua estrutura não recorrente e à carência de algoritmos eficientes de aprendizagem para estruturas neurais com realimentação. Embora análise e síntese de redes neurais recorrentes sejam mais complexas, estes modelos têm proporcionado desempenho superior em diversas aplicações. Devido à sua estrutura, redes recorrentes são mais difíceis de serem analisadas, o processo de treinamento é mais trabalhoso, e os algoritmos de aprendizagem mais complexos e lentos [1]. A incorporação de recorrência em sistemas neurais nebulosos faz com que estes sistemas sejam capazes de tratar problemas que envolvam relações temporais, já que a presença de realimentação permite a criação de representações internas e os mecanismos de memória necessários para processar e armazenar este tipo de informação.

Este trabalho analisa e compara o efeito da recorrência em uma classe de redes neurais nebulosas, partindo-se de uma estrutura neural estática como base para duas estruturas multicamadas recorrentes. Dois tipos de recorrência são analisados: recorrência interna local e recorrência interna global. Estas três estruturas têm duas partes em comum. A primeira parte constitui um sistema de inferência nebuloso para o processamento dos sinais de entrada. Além de neurônios que fornecem o grau de pertinência das entradas a conjuntos nebulosos que granularizam os respectivos universos na primeira camada, neurônios lógicos do tipo *AND* [2], compõem a segunda camada. As relações temporais são induzidas por uma realimentação local e global nos neurônios lógicos desta camada intermediária, fornecendo assim, elementos de memória necessários para a representação temporal. A segunda parte é constituída por neurônios não lineares clássicos organizados em uma camada que atua como uma função não linear de agregação [3].

Os pesos da camada de saída tanto da rede neural nebulosa estática, como das estruturas recorrentes analisadas, são ajustados utilizando o gradiente do erro quadrático. Os pesos da camada intermediária das três estruturas são ajustados utilizando um método de treinamento por reforço associativo. Neste caso, os pesos da camada intermediária são ajustados usando um mecanismo de recompensa e punição inspirado em [4] e [5].

Neste trabalho, as estruturas neural nebulosa estática **RNFest**, neural nebulosa recorrente local **RNFRLoc** e neural nebulosa recorrente global **RNFRGlob** são analisadas comparando o desempenho com as correspondentes estruturas de redes neurais

multicamada MLP [6], recorrente RNRLoc_Cheng [7], e nebulosa recorrente RNFRLoc_Lee [1]. A comparação é feita considerando dois sinais de entrada em um problema de modelagem de um sistema dinâmico não linear. O artigo está organizado da seguinte forma. A próxima seção descreve a estrutura da rede neural nebulosa proposta; a seção 3 apresenta o algoritmo de aprendizado. A simulação de um sistema dinâmico não linear e a comparação dos resultados com outras metodologias são apresentadas na seção 4. Finalmente, na seção 5 discute-se as conclusões do trabalho.

2. Estruturas de Redes Neurais Nebulosas

Esta seção detalha as estruturas de redes neurais nebulosas de interesse neste trabalho. Estas estruturas são compostas por duas partes: um sistema de inferência nebuloso e uma rede neural clássica. A estrutura da rede neural nebulosa com recorrência global é apresentada na Figura 1. Note que a recorrência ocorre nos neurônios da camada intermediária. As redes neurais com recorrência parcial ou local têm estrutura análoga, porém com realimentação na saída dos neurônios da segunda camada. A rede estática correspondente possui a mesma estrutura, a menos das conexões laterais e de realimentação.

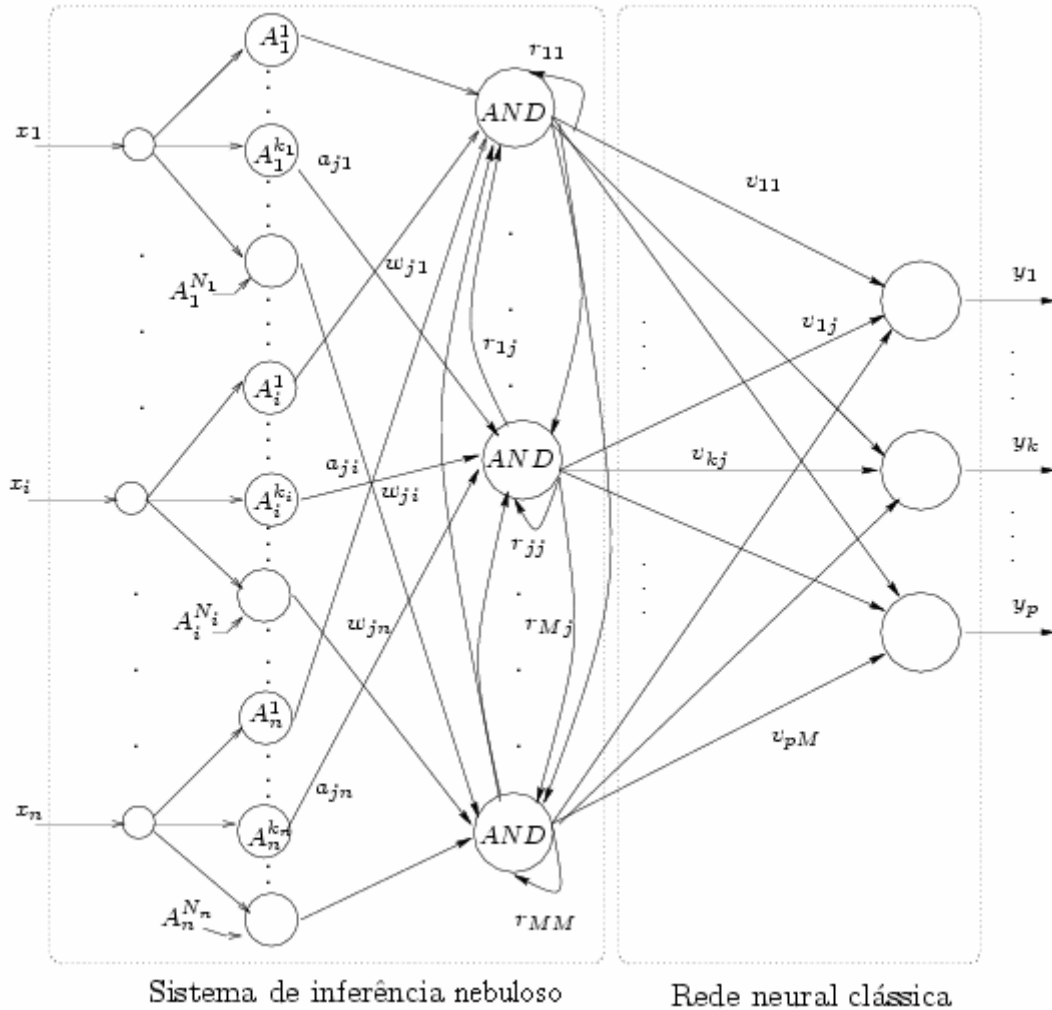


Figura 1: Rede neural nebulosa com recorrência global.

A segunda camada é constituída por neurônios lógicos do tipo AND (Figura 2), cujo processamento se dá através de operadores da classe das *t-normas* e *s-normas* [2]. Já a camada de saída é formada por neurônios não lineares clássicos. Seu propósito é realizar uma combinação das saídas dos neurônios da camada anterior via uma função de agregação não linear. O sistema de inferência é composto pelas camadas de entrada e intermediária. A camada de entrada contém neurônios cujas funções de ativação são as funções de pertinência dos conjuntos nebulosos que formam a partição do espaço de entrada. Isto é, para cada dimensão $x_i(t)$ do vetor de entrada n -dimensional $\mathbf{x}(t)$ existem N_i conjuntos nebulosos $A_i^{k_i}$, $k_i = 1, \dots, N_i$ cujas funções de pertinência são as correspondentes funções de ativação dos neurônios da camada de entrada. A variável t denota o tempo discretizado, isto é, $t = 1, 2, \dots$ e será omitida no decorrer do artigo para simplificar a notação. Deste modo, os graus de pertinência associados aos padrões de entrada são calculados da seguinte forma:

$$a_{ji} = \mu_{A_i^{k_i}}(x_i) \quad (1)$$

para $i = 1, \dots, n$ e $j = 1, \dots, M$; onde M é o número de neurônios da camada intermediária.

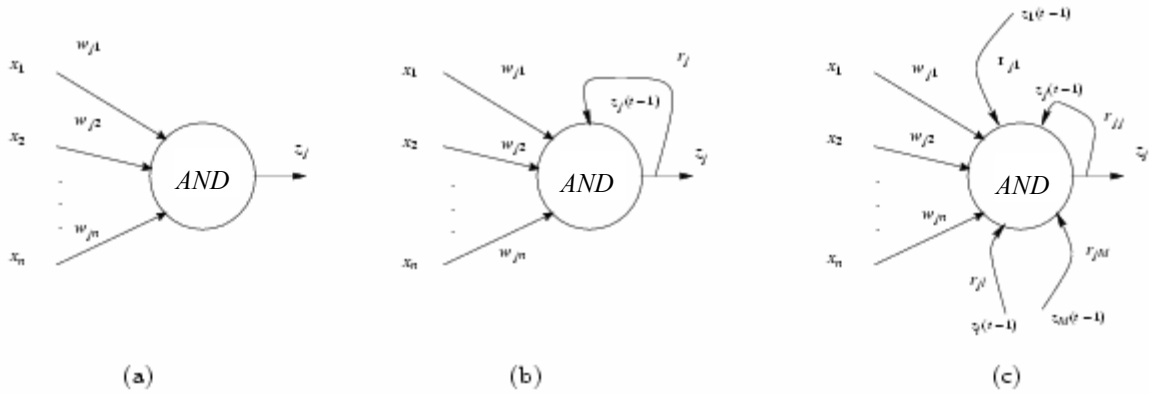


Figura 2: Neurônio lógico AND. (a) estático, (b) recorrência interna local, (c) recorrência interna global.

A saída z_j de um neurônio da camada intermediária é definida por:

$$z_j = \mathbf{T} \left(\sum_{i=1}^{n+D} w_{ji} s_{ji} \right) \quad (2)$$

Note que para $D = 0$ obtém-se as saídas z_j da rede **RNFest** enquanto que para $D = 1$, as saídas z_j são aquelas da rede **RNFRLoc**, fazendo $w_{jn+1} = r_{jj} = r_j$ e $a_{jn+1} = q^{-1} z_j$. Para $D = M$ obtém-se as saídas z_j da rede **RNFRGlob**. Neste caso, $w_{jn+l} = r_{jl}$, e $a_{jn+l} = q^{-l} z_j$, $l = 1, \dots, M$, onde q^{-l} é o operador de atraso.

É importante verificar que cada uma das estruturas analisadas gera um conjunto de regras do tipo SE-ENTÃO $R = \{R_j, j = 1, \dots, M\}$. Por exemplo, para o caso da rede **RNFest** estas tem a seguinte forma:

$$R_j: \text{SE } (x_1 \text{ é } A_i^{k1} \text{ com certeza } w_{j1}) \dots \text{AND } (x_i \text{ é } A_i^{ki} \text{ com certeza } w_{ji}) \dots \text{AND } (x_n \text{ é } A_n^{kn} \text{ com certeza } w_{jn}) \dots \text{ENTÃO } z \text{ é } z_j$$

Isto significa que existe uma dualidade entre o sistema de inferência nebuloso neural e um sistema baseado em regras nebulosas. No caso das estruturas recorrentes, as regras R_j incorporam as suas próprias saídas através do operador de atraso q^{-l} , atuando como variáveis internas de um sistema de inferência nebuloso recorrente equivalente [8].

A segunda parte da estrutura é uma rede neural clássica, composta por neurônios não lineares e não recorrentes. A saída y_k corresponde à agregação dos valores de z_j e v_{kj} , $j = 1, \dots, M$ e $k = 1, \dots, p$, isto é:

$$y_k = \psi(u) = \psi \left(\sum_{j=1}^M (v_{kj} z_j) \right) \quad (3)$$

onde $\psi: \mathfrak{R}^n \rightarrow [0,1]$ é uma função monotonicamente crescente. Neste trabalho, considera-se a função sigmóide: $\psi(u) = 1/(1 + \exp(-u))$.

A dinâmica da rede neural nebulosa pode ser resumida da seguinte maneira:

1. N_i é o número de conjuntos nebulosos que constitui a partição da i -ésima entrada;
2. O índice j indexa os neurônios **AND**. Para as estruturas neurais nebulosas analisadas, j é determinado utilizando a seguinte expressão:

$$j = f(K) = k_n + \sum_{i=2}^n (k_{(n-i+1)} - 1) \left(\prod_{\tau=1}^{i-1} N_{(n+1-\tau)} \right) \quad (4)$$

onde, $K = (k_1, \dots, k_i, \dots, k_n)$, sendo k_i o índice associado à regra ativada pelo componente x_i do padrão de entrada \mathbf{x} ;

3. $a_{ji} = \mu_{A_i^{ki}}(x_i)$ é o grau de pertinência de x_i no conjunto nebuloso A_i^{ki} , sendo a_{ji} a entrada para o neurônio j da camada intermediária;
4. z_j é a j -ésima saída da camada intermediária definida pela Equação (2);
5. y_k é a k -ésima saída da rede conforme a Equação (3);
6. w_{ji} é o peso entre o j -ésimo neurônio **AND** e o i -ésimo neurônio da camada de entrada;
7. v_{kj} é o peso entre a saída y_k da rede e o j -ésimo neurônio **AND**;

8. No caso da rede **RNFRLoc** $r_j = r_{jj}$ é o peso da recorrência e no caso da rede **RNFRGlob** r_{jl} são os pesos das conexões laterais entre o j -ésimo neurônio *AND* e o l -ésimo neurônio da segunda camada.

As arquiteturas apresentadas têm como vantagens a geração automática da topologia da rede durante a aprendizagem, flexibilidade quanto à utilização de normas triangulares e a possibilidade de extrair regras nebulosas diretamente da topologia [5] e [9].

3. Método de Aprendizagem

Assim como as redes neurais clássicas, as redes neurais nebulosas recorrente e estática apresentadas possuem capacidade de aprendizado. O algoritmo de aprendizado proposto está baseado em dois métodos: o método do gradiente da função erro [10], [11] para os pesos da camada de saída e o método de aprendizado por reforço associativo [4], [5] para os pesos da camada intermediária.

Os passos do algoritmo de treinamento são enumerados a seguir e detalhados nas sub-seções seguintes.

1. Gerar as funções de pertinência;
2. Inicializar os pesos;
3. Até que a condição de parada não seja satisfeita, fazer:
 - 3.1 Apresentar um padrão \mathbf{x} à rede, geralmente escolhido aleatoriamente;
 - 3.2 Efetuar a fuzzificação;
 - 3.3 Determinar os neurônios *AND* ativos e calcular a saída da rede \mathbf{y} ;
 - 3.4 Atualizar os pesos w_{ji} , v_{kj} e r_{jl} para a rede com recorrência global **RNFRGlob**, r_j para a rede com recorrência local **RNFRLoc**;
 - 3.5 Testar a condição de parada (máximo erro permitido ou número máximo de iterações).

3.1 Geração das funções de pertinência

As funções de pertinência consideradas para os três modelos neurais nebulosos são funções triangulares, podendo ser adotadas, em geral, outras formas (trapezoidais, gaussianas, etc). Estas funções de pertinência, normais e complementares, são caracterizadas por três parâmetros x_{imin} , x_{imax} e c_{ir} , sendo x_{imin} e x_{imax} os valores mínimo e máximo associados ao i -ésimo componente do vetor de entrada \mathbf{x} e c_{ir} , o centro da função de pertinência, com $r = 1, \dots, N_i$, onde N_i é o número de conjuntos nebulosos que compõem a partição do universo do i -ésimo componente de \mathbf{x} .

Dois tipos de partições podem ser utilizadas: partição uniforme (Figura 3) e partição não uniforme (Figura 4).

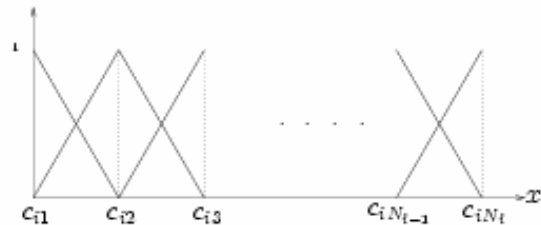


Figura 3: Partição uniforme.

Dado o valor de N_i , a partição uniforme é gerada da seguinte forma:

$$\Delta_i = \frac{x_{imax} - x_{imin}}{N_i - 1} \quad (5)$$

para $i = 1, \dots, n$. Os centros c_{ir} são definidos como:

$$c_{ir} = x_{imin} + (r - 1)\Delta_i \quad (6)$$

para $r = 1, \dots, N_i$.

Quando ocorre uma concentração de dados em uma determinada região do espaço de entrada, pode ser interessante assumir uma partição não uniforme, como ilustra a Figura 4, resultando em um menor número de centros em regiões onde a concentração de informação seja baixa [5].

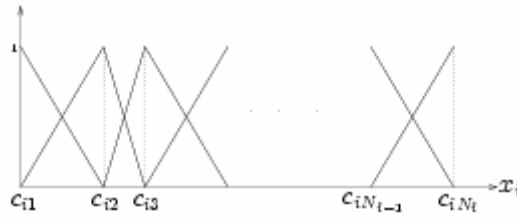


Figura 4: Partição não uniforme.

Neste caso, técnicas de agrupamento para determinar os centros c_{ir} , com $r = 1, \dots, N_i$ são necessárias. O algoritmo de agrupamento utilizado neste trabalho é baseado na rede neural auto-organizada de Kohonen [12], com algoritmo de treinamento LVQ (*Learning Vector Quantization*), proposto em [13].

Uma vez determinados os centros, os graus de pertinência $\mu_{Ai}^{k_i}(x_i)$ podem ser determinados para cada padrão de entrada x .

3.2 Inicialização dos pesos

Os pesos da camada de saída v_{kj} são inicializados aleatoriamente com valores no intervalo $[-1, 1]$, $k = 1, \dots, p$ e $j = 1, \dots, M$. Os pesos da recorrência r_{jl} para a rede **RNFRLoc** e r_{jl} para a rede **RNFRGlob**, assim como, os pesos w_{ji} , são inicializados com valores aleatórios no intervalo $[0, 1]$, para $i = 1, \dots, n$, $j = 1, \dots, M$ e $l = 1, \dots, M$.

3.3 Determinação dos neurônios AND ativos

Para cada padrão de entrada x apresentado à rede **RNFRGlob**, **RNFRLoc** ou **RNFEst**, existe no máximo dois conjuntos nebulosos com graus de pertinência diferentes de zero, denominados conjuntos ativos que, por sua vez, definem os neurônios AND ativos. Assim, de M neurônios AND na segunda camada, no máximo 2^n estarão ativos, sendo estes, identificados da seguinte maneira:

Dado um padrão de entrada $x = [x_1, \dots, x_i, \dots, x_n]$ e $K^l = (k_1^l, \dots, k_i^l, \dots, k_n^l)$ o vetor cujos componentes são os índices da primeira função de pertinência diferente de zero para cada componente de x . Seja $K^2 = (k_1^2, \dots, k_i^2, \dots, k_n^2)$ um vetor tal que:

$$k_i^2 = \begin{cases} k_i^1 + 1 & \text{se } \mu_{A_i^{k_i^1}}(x_i) \neq 1 \\ k_i^1 & \text{caso contrário} \end{cases}$$

O número de neurônios AND ativos N_a é 2^{Pa} sendo $2^{Pa} \leq 2^n$, onde Pa é o número de elementos tal que $k_i^1 \neq k_i^2$, para $i = 1, \dots, n$. Assim, os neurônios AND ativos são determinados utilizando a Equação (4).

3.4 Fuzzificação

Nesta etapa, os graus de pertinência para os subconjuntos ativos definidos em K^l para os padrões de entrada são calculados. No caso em que $k_i^1 \neq k_i^2$, tem-se:

$$\mu_{A_i^{k_2}}(x_i) = 1 - \mu_{A_i^{k_1}}(x_i) \tag{7}$$

isto devido à complementaridade das funções de pertinência. Desta forma, o cálculo é efetuado a apenas $2n$ graus de pertinência.

3.5 Atualização dos pesos

O processo de atualização dos pesos das conexões das redes neurais nebulosas propostas é baseado no método do gradiente da função erro e no método de treinamento por reforço associativo proposto em [4], [13]. O gradiente do erro quadrático é utilizado para atualizar os pesos v_{kj} da rede neural clássica. Os pesos das conexões que formam o sistema de inferência nebuloso são atualizados através do método de treinamento por reforço associativo.

Uma vez avaliada as saídas de cada neurônio para um determinado padrão de entrada, utilizando as expressões (2) e (3), o objetivo do processo de treinamento supervisionado é minimizar o erro quadrático médio entre a saída atual da rede e a saída desejada, isto é, minimizar:

$$\varepsilon = \frac{1}{p} \sum_{k=1}^p (y_k - \hat{y}_k)^2 \quad (8)$$

onde, \hat{y}_k é o valor da k -ésima saída da rede no instante t e y_k é a k -ésima saída desejada para o correspondente padrão x . Assim, se v_{kj} é um peso conectado a k -ésima unidade de saída, tem-se:

$$\Delta v_{kj} = \eta (y_k - \hat{y}_k) \psi'(u) z_j \quad (9)$$

onde $\psi'(u) = \psi(u)(1 - \psi(u))$ é a derivada da função de ativação da camada de saída avaliada em u , e η é a taxa de aprendizagem.

Os pesos da camada intermediária são atualizados através de um mecanismo de recompensa e punição, inicialmente proposto em [14]. Este mecanismo utiliza um sinal de reforço δ , que em [4], é proposto como sendo $\delta = 1 - \varepsilon$, onde ε é o erro médio (Equação 8). Nota-se que grandes valores de δ correspondem a uma melhor aproximação entre a saída da rede e o sinal desejado e os pesos são aumentados. Já valores baixos de δ apontam um desempenho pobre da rede, fornecendo um erro alto e indicando que as regras ativadas não foram corretas, tendo então que diminuir os graus de certeza destas regras. Desta forma, a regra de atualização dos pesos da camada intermediária são atualizados como segue:

$$\Delta w_{ji} = \delta \alpha_1 (1 - w_{ji}) - (1 - \delta) \alpha_2 w_{ji} \quad (10)$$

$$\Delta r_{ji} = \delta \alpha_3 (1 - r_{ji}) - (1 - \delta) \alpha_4 r_{ji} \quad (11)$$

onde, $0 < \alpha_1 \ll \alpha_2 < 1$ e $0 < \alpha_3 \ll \alpha_4 < 1$ são as taxas de aprendizagem, $j = 1, \dots, M$ e $i = 1, \dots, n$.

4. Simulações e Comparações

Esta seção apresenta os resultados de simulação considerando um problema de identificação de um sistema dinâmico não linear. O exemplo adotado, sugerido em [6], é um sistema não-linear, descrito pela equação a diferenças de terceira ordem:

$$y(t+1) = f[y(t), y(t-1), y(t-2), u(t), u(t-1)] \quad (12)$$

onde $u(t)$ é a entrada ao sistema e a função $f[\cdot]$, suposta desconhecida, tem a forma:

$$f[x_1, x_2, x_3, x_4, x_5] = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_3^2 + x_2^2} \quad (13)$$

O conjunto de treinamento foi composto de 2400 padrões, dentre os quais os 800 primeiros foram gerados considerando um sinal de entrada $u(t)$ aleatório com distribuição uniforme, no intervalo $[-1, 1]$. Os outros padrões foram gerados utilizando a função $u(t) = \sin(2\pi t / 250)$, a modo de excitar todos os modos do sistema. As condições iniciais tanto para o conjunto de treinamento como para os conjuntos de teste foram $y(0) = y(-1) = y(-2) = u(1) = 0$.

As entradas utilizadas para cada rede são descritas na Tabela 1. A Tabela 2 caracteriza os parâmetros que definem a complexidade das estruturas avaliadas, assim como o número de parâmetros ativos por iteração. Ainda na Tabela 2, n refere-se ao número de entradas do modelo neural, p é o número de saídas, N_i é o número de conjuntos nebulosos que formam a partição para cada entrada x_i da rede, $N_{a_{max}}$ é o número máximo de neurônios da camada intermediária ativos para cada padrão apresentado. Parâmetros ativos são aqueles correspondentes aos neurônios ativos.

Tabela 1: Entradas das redes neurais.

Estrutura	n	Entradas utilizadas
RNFEst	5	$y(t), y(t-1), y(t-2), u(t), u(t-1)$
RNFRGlob	1	$u(t)$
RNFRLoc	1	$u(t)$
RNRLoc_Cheng	1	$u(t)$
RNFRLoc_Lee	2	$y(t), u(t)$
MLP	5	$y(t), y(t-1), y(t-2), u(t), u(t-1)$

Tabela 2: Características das redes neurais.

Estrutura	n	p	Tipo de Agrupamento	N_i	Na_{max}	Parâmetros / iteração	Parâmetros
RNFEst	5	1	UNIFORME	[7,7,7,6,6]	32	$Na_{max}(n+p)$	192
RNFRGlob	1	1	UNIFORME	[30]	2	$Na_{max}(n+p+1)$	8
RNFRLoc	1	1	UNIFORME	[30]	2	$Na_{max}(n+p+Na_{max})$	6
RNRLoc_Cheng ¹	1	1	---	[30]	50	$Na_{max}(n+p+2)+p$	201
RNFRLoc_Lee	2	1	---	[30,30]	30	$3n Na_{max} + Na_{max} p$	210
MLP ²	5	1	---	[30,20]	[30,20]	$M_1(n+1) + M_2(M_1+1) + (M_2+1)p$	821

¹ Para RNRLoc_Cheng, $N_i = Na_{max}$ é o número de neurônios da camada intermediária.

² Para MLP, $N_i = Na_{max} = [M_1, M_2]$, sendo M_1 e M_2 o número de neurônios das camadas intermediárias.

As taxas de aprendizado para os modelos propostos foram $\eta = 0,5$ para os pesos da camada de saída e para os pesos da camada intermediária e de recorrência as taxas com valores $\alpha_1 = \alpha_3 = 0,001$ e $\alpha_2 = \alpha_4 = 0,0001$. Os modelos de redes neurais nebulosos propostos utilizaram como s -norma a soma probabilística e como t -norma o produto algébrico.

Uma vez ajustados os parâmetros dos modelos, estes foram testados com dois conjuntos de testes diferentes, ambos compostos por 1000 pontos, gerados utilizando sinais de entrada $u(t)$ diferentes. O primeiro conjunto de teste foi gerado considerando um sinal de entrada $u(t)$ definido pela equação:

$$u(t) = u_1(t) = 0.8 \text{sen}(2\pi t / 250) + 0.2 \text{sen}(2\pi t / 25) \quad (14)$$

O segundo conjunto de teste foi gerado utilizando um sinal de entrada definido como:

$$u(t) = u_2(t) = \begin{cases} \text{sen}(\pi t / 25) & t \leq 250, \\ 0.2 & 250 < t \leq 500, \\ -0.2 & 500 < t \leq 700, \\ 0.3 \text{sen}(\pi t / 25) + 0.1 \text{sen}(\pi t / 32) + 0.6 \text{sen}(\pi t / 10) & t > 700 \end{cases} \quad (15)$$

A Tabela 3 mostra os erros quadráticos médios (EQM) de treinamento, os erros de teste e os tempos de processamento (T_{proc}) para o treinamento de cada uma das redes consideradas neste trabalho.

Tabela 3: Erros quadráticos médios (EQM) para o exemplo de identificação.

Estrutura	Épocas	T_{proc} (s)	EQM Treinamento ($\times 10^{-3}$)	EQM Teste 1 ($\times 10^{-2}$)	EQM Teste 2 ($\times 10^{-2}$)
RNFEst	31	24	0,000	0,048	0,562
RNFRGlob	800	222	0,000	0,036	0,070
RNFRLoc	800	193	0,000	0,036	0,067
RNFLoc_Cheng	600	256	0,002	0,059	0,104
RNFRLoc_Lee	500	983	0,008	0,130	0,170
MLP	500	1021	0,002	0,021	0,027

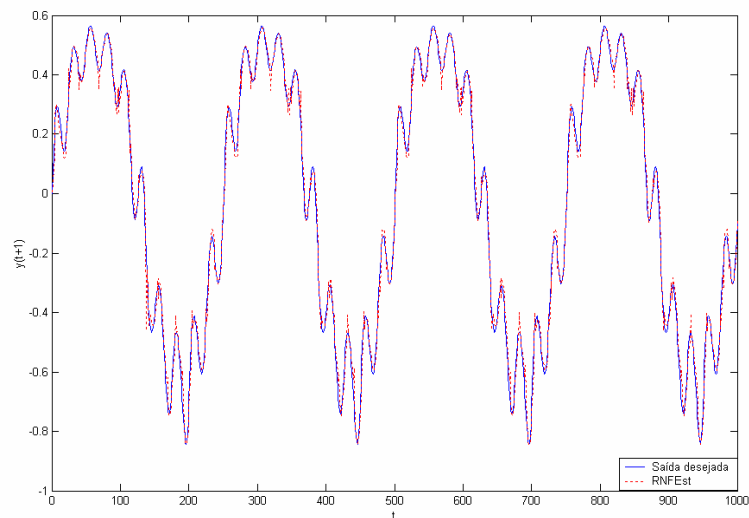
A análise da Tabela 2 e Tabela 3 sugere que, em termos de um compromisso entre complexidade e desempenho, as redes neurais nebulosas recorrentes propostas são tão ou mais eficientes que as redes correspondentes **RNFRLoc_Lee**, **RNRLoc_Cheng** e **MLP**.

Por exemplo, pode-se notar que a rede **RNFRLoc_Lee** [1] utiliza duas entradas, 30 conjuntos nebulosos em cada partição destas entradas, significando um número maior de parâmetros a serem ajustados por iteração. Além disso, o EQM de teste (Tabela 1) foi maior que o EQM de teste das redes **RNFEst**, **RNFRGlob** e **RNFRLoc**. Por outro lado, embora o EQM de teste obtido pela rede **MLP** seja menor que os obtidos pelas redes propostas, cabe considerar que o tempo de processamento desta rede foi maior, assim como o número de parâmetros e o número de entradas (cinco, conforme sugere a expressão (12) e Tabela 2). Claramente, o número de parâmetros no caso das redes neurais nebulosas propostas é substancialmente menor.

A recorrência aplicada nas redes neurais nebulosas apresentou-se como uma opção vantajosa pois a necessidade de informação para identificar os sistemas foi menor, isto em relação ao número de entradas para modelar os sistemas, além do período de treinamento utilizado.

Resumindo, as redes neurais nebulosas analisadas neste trabalho produzem modelos mais simples e parcimoniosos pois contém um número reduzido de entradas e de parâmetros, requer um tempo baixo de aprendizagem, além de proporcionar menores erros de aproximação do que redes com estruturas similares sugeridas na literatura. Isto significa maior eficiência computacional em termos de exigências de memória e tempo de processamento, além de modelos mais transparentes e com erros aceitáveis de aproximação.

Cabe observar que os resultados obtidos para as redes neurais nebulosas recorrentes propostas foram similares. Isto devido à restrição na quantidade de neurônios lógicos ativos por iteração na camada intermediária, indicando que nem sempre é necessária uma rede neural mais complexa para obter os melhores resultados. Assim, é necessário o estabelecimento de um compromisso entre complexidade da solução e desempenho desejado. A Figura 5 e a Figura 6 ilustram as saídas das redes neurais nebulosas propostas, para o primeiro e segundo teste, respectivamente.



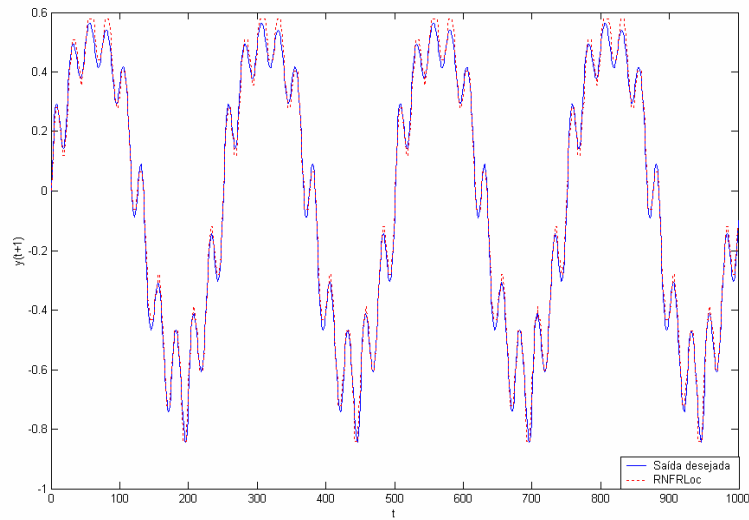
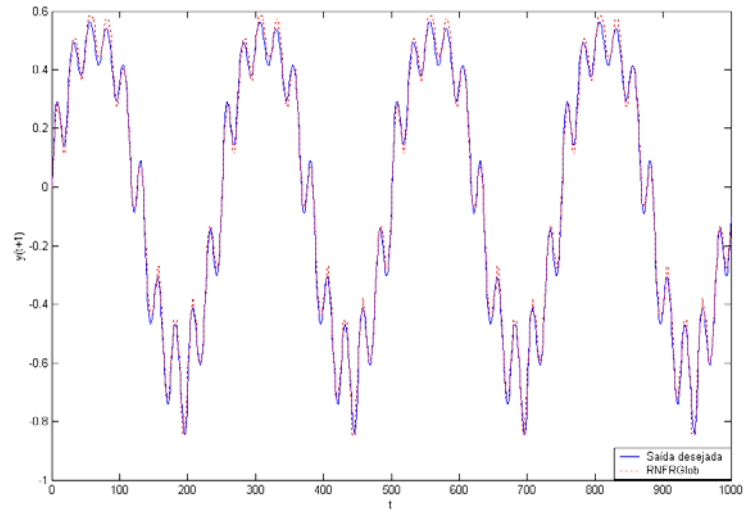
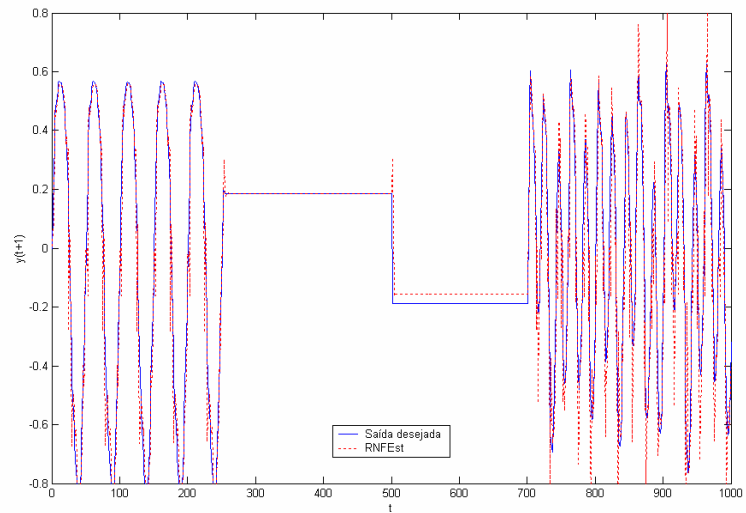


Figura 5: Saídas das redes neurais nebulosas propostas para o primeiro teste (utilizando $u_1(t)$).



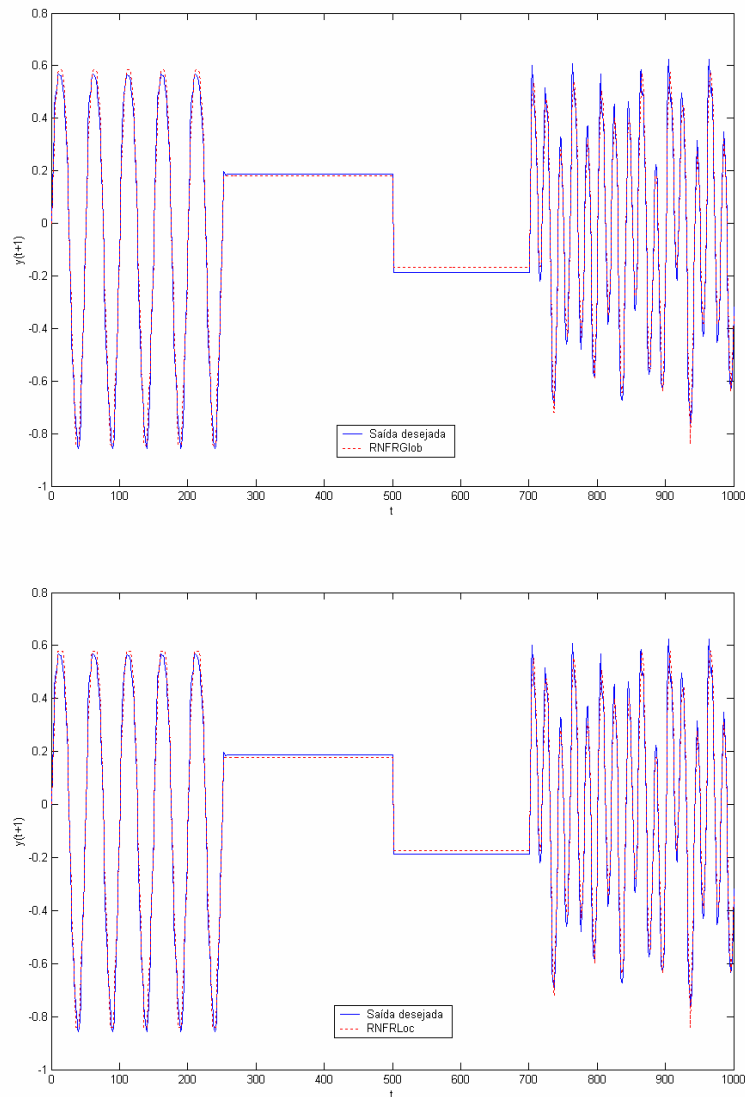


Figura 6: Saídas das redes neurais nebulosas propostas para o segundo teste (utilizando $u_2(t)$).

5. Conclusões

Neste trabalho três diferentes estruturas de redes neurais nebulosas foram consideradas. As estruturas contêm um sistema de inferência nebuloso composto por neurônios lógicos que utilizam normas triangulares para o processamento da informação. O sistema de inferência nebuloso está conectado em série com uma rede neural clássica.

O algoritmo de aprendizado proposto para os modelos neurais nebulosos apresentados neste trabalho mostrou-se simples, rápido e não necessita de derivadas para o ajuste dos pesos da camada intermediária e da recorrência.

Os resultados de simulação demonstraram que os modelos neurais nebulosos apresentados neste trabalho são capazes de modelar sistemas com dinâmica não linear, reduzindo a quantidade de informação *a priori* necessária para a resolução de problemas de modelagem.

Cabe observar que, embora, os modelos neurais nebulosos tenham apresentado um menor tempo de processamento e necessidade de recursos por iteração para atingir um desempenho igual ou melhor aos outros modelos testados, estas estruturas são de complexidade exponencial quanto a granularidade dos universos e o número das variáveis de entrada.

Experimentos computacionais, considerando um problema de modelagem de processo não linear, sugerem que, quando comparadas com estruturas similares propostas na literatura, as redes neurais nebulosas são eficazes para aproximar modelos de sistemas dinâmicos não lineares com erros de aproximação satisfatórios.

6. Agradecimentos

Os autores gostariam de agradecer aos revisores anônimos pelas sugestões. Os autores agradecem à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) processos #300729/86-3 e #132544/2001-6, pelo auxílio.

7. Referências

- [1] C.H. Lee e C.C. Teng: “Identification and control of dynamic systems using recurrent fuzzy neural networks”, IEEE Transactions on Fuzzy Systems, vol. 4, no 8, 2000, pp. 349 – 366.
- [2] W. Pedrycz e F. Gomide: “An introduction to fuzzy sets: Analysis and design”, MIT Press, Cambridge, MA, 1998.
- [3] I. Luna, R. Ballini e F. Gomide: “Rede neurofuzzy recorrente para identificação e controle de sistemas dinâmicos discretos”, Anais do XIV Congresso Brasileiro de Automação, Natal-Brasil, Setembro 2000, pp. 353-358.
- [4] A. Barto e M. Jordan: “Gradient following without backpropagation in layered networks”, Proceedings of the IEEE First International Conference on Neural Networks, no. 2, 1987, pp. 629 – 636.
- [5] W. Caminhas, H. Tavares, F. Gomide e W. Pedrycz: “Fuzzy sets based neural networks: structure, learning and applications”, Journal of Advanced Computational Intelligence, vol. 3, no. 3, 1999, pp. 151 – 157.
- [6] K. Narendra e K. Parthasarathy: “Identification and control of dynamical systems using neural networks”, IEEE Transactions on Neural Networks, vol. 1, no. 1, 1990, pp. 4 – 27.
- [7] Y. Cheng, T. Karjala e D. Himmelblau: “Closed loop nonlinear process identification using internally recurrent nets”, Neural Networks, vol. 10, no. 3, 1997, pp. 573 – 586.
- [8] H. Bersini e B. Gorrini: “Recurrent fuzzy systems”, Technical Report TR/IRIDIA/94-11, Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle, 1994.
- [9] E. Iyoda: “Inteligência computacional no projeto automático de redes neurais híbridas e redes neurofuzzy heterogêneas”, Tese de Mestrado, FEEC – Unicamp, Brasil, Janeiro 2000.
- [10] D.R. Hush e B.G. Horne: “Progress in supervised neural networks”, IEEE Signal Processing Magazine, January 1993, pp. 8 – 39.
- [11] D.E. Rumelhart e J.L. McClelland: “Parallel Distributed Processing”, MIT Press, Cambridge, vol. 1, 1986.
- [12] T. Kohonen: “Self-Organized Formation of Topologically Correct Feature Maps”, Biological Cybernetics, vol. 43, 1982, pp. 59 – 69.
- [13] W. Caminhas: “Estratégias de Detecção e Diagnóstico de Falhas em Sistemas Dinâmicos”, Tese de doutorado, FEEC – Unicamp, 1997.
- [14] R. Ballini e F. Gomide: “Learning in recurrent hybrid neurofuzzy networks”, IEEE International Conference on Fuzzy Systems, 2002, pp. 785 – 791.