Redes Neurais Locais-Globais Uma Aplicação ao Problema de Dados Faltantes

Carlos E. Pedreira[1], Luiz Carlos Pedroza[2] e Mayte Fariñas[1]

[1] DEE PUC-RIO CP. 38063 Rio de Janeiro, CEP 22452-970
[2] CEFET-RJ Av. Maracanã, 229 Rio de Janeiro, CEP 20271-110
E-mails: pedreira@ele.puc-rio.br, pedroza@cefet-rj.br, mayte@ele.puc-rio.br.

Abstract:

In this paper a new connectionist architecture is proposed. The proposed architecture is trained by a scheme based on partition of the function domain, approximating the generator function by a set of very simple supporting functions. This method has an interesting ability concerning interpolation. A synthetic experiment and a real data missing data application are presented.

1. Introdução

Muitos problemas de interesse prático estão relacionados a interpolação dos dados. Entre estes, de particular relevância está o problema de preenchimento de dados faltantes. Neste caso, o que se procura é uma forma de emular uma função em um intervalo do domínio onde apenas uma parte dos pontos é conhecida. A preocupação aqui não é a capacidade preditiva da rede, isto é, estimar pontos fora do domínio, mas sim de reproduzir a função da melhor forma possível dentro do domínio estabelecido. Neste artigo, propõe-se um algoritmo capaz de reconstruir a função a partir de estimativas locais ao longo do domínio de interesse.

A arquitetura conexionista proposta é não usual e utiliza uma metodologia de treinamento baseada no particionamento do domínio da função a ser emulada. Este função é aproximada, por um conjunto de funções de apoio muito simples, muitas vezes lineares. A idéia central é expressar o mapeamento entrada-saída através de uma função composta por partes. A estrutura básica é constituída pela combinação de vários pares compostos de funções de aproximação e funções de pertinência. As funções de pertinência definem em cada trecho do domínio a participação da função de ativação a essa associada. É possível a ocorrência de sobreposições parciais das funções de pertinência proporcionando uma maior riqueza do mapeamento pretendido. Desse modo o problema de aproximação de funções é enfocado especializando-se grupos de neurônios, formados pelos pares anteriormente descritos, que emulam a função geradora em cada setor do domínio. O grau de especialização em um determinado trecho é dado pelo nível da função de pertinência. Por exemplo, em um trecho aonde apenas uma das funções de pertinência assume valor alto, haverá uma dominância da função de aproximação associada a esta.

2. A arquitetura Proposta

Consideremos uma rede com m nós ou neurônios. Seja $\{x_i\}_1^n$ a partição dos dados usada para treinamento. Por simplicidade algébrica e de notação iremos considerar o caso onde $x \in \acute{}$ (o subscrito de x será omitido), a generalização para o caso onde $x \in \acute{}^{\,n}$ é algebricamente direta. Define-se, para cada ponto x, m funções de pertinência do seguinte modo:

$$B_j(x) = -C_j\left[\frac{1}{1+\exp(d_j(x-h_j^{(1)}))} - \frac{1}{1+\exp(d_j(x-h_j^{(2)}))}\right], \quad j=1,\ldots,m$$

onde C_j, d_j, $h_j^{(1)}$ e $h_j^{(2)}$ são parâmetros a serem ajustados. Note-se que o parâmetro C_j reflete o nível da função de pertinência, enquanto d_j está relacionado à declividade desta função. Os parâmetros $h_j^{(1)}$ e $h_j^{(2)}$ delimitam o setor do domínio no qual a função de aproximação associada a esta função de pertinência é mais ativa. (ver figura 1).

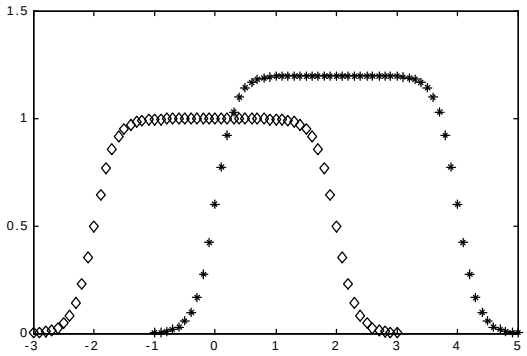


Figura 1 – Exemplos de funções de ativação
◊ C=1, d=6; $h_j^{(1)}$ = -2; $h_j^{(2)}$ =2
* C=1.2, d=6; $h_j^{(1)}$ = 0; $h_j^{(2)}$ =4

As funções de aproximação são tipicamente funções lineares ou quadráticas. Embora funções mais complexas possam ser usadas sem prejuízo da estrutura

teórica proposta, não parecem trazer contribuição significativa ao modelo. Consideremos funções de aproximação lineares:

$$\kappa_j(x) = a_j x + b_j \qquad j=1, \ldots, m$$

onde a_j e b_j são os parâmetros a serem estimados. Cada nó, ou neurônio, da rede é constituído de um par { função de pertinência ; função de aproximação} (ver figura 2). Então, para cada nó é necessário estimar 6 parâmetros (7 no caso de funções de aproximação quadráticas). Como usualmente, o número de nós indica a complexidade do modelo.

As entradas são conectadas ao nó onde é efetuado o produto da função de pertinência $B_j(x)$ e da função de aproximação $\kappa_j(x)$. A saída da rede é um somatório da saída de cada um destes nós. Note que não há pesos ligando a saída dos nós a saída da rede (veja figura 2). Deste modo a saída do j-ésimo nó é $B_j(x)\ \kappa_j(x)$, e a saída da rede é dada por:

$$g^m(x) = \sum_{j=1}^{m} B_j(x)\kappa_j(x) \qquad (2.1)$$

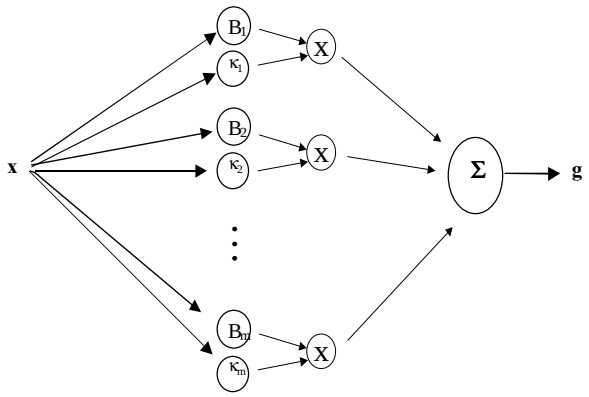


Figura 2 A arquitetura proposta

Definindo-se $B_{vet}(x) \equiv (B_1(x), B_2(x), \ldots B_m(x))$ e $\kappa_{vet}(x) \equiv (\kappa_1(x), \kappa_2(x), \ldots \kappa_m(x))$ pode-se escrever a saída da rede em forma de produto interno, i.e. $g^m(x) = < B_{vet}(x), \kappa_{vet}(x) >$.

O objetivo central é projetar uma rede cuja saída aproxime uma função alvo, f(x), da melhor forma possível. Define-se então, uma função de erro como uma combinação convexa do quadrado de duas medidas de erro E_1 e E_2:

$$E \equiv a \sum_{i=1}^{k} E_1^2(x_i) + (1-a)\, E_2^2 \qquad (2.2)$$

onde

$$E_1(x_i) \equiv g^m(x_i) - y(x_i) \quad \text{e} \quad E_2 \equiv 1 - \sum_{j=1}^{m} C_j \qquad (2.3)$$

O termo E_1 está associado a qualidade da aproximação obtida enquanto E_2 é usado com a finalidade de manter as funções de pertinência limitadas. Na realidade se está penalizando soluções nas quais o somatório destas funções excede 1. A escolha da unidade como valor limitante não é necessária embora confira mais interpretabilidade aos resultados.

Define-se, para cada neurônio, um vetor de parâmetros $\Im^j \equiv (C_j, d_j, h_j^{(1)}, h_j^{(2)}, a_j, b_j)$ e o objetivo central será encontrar $\Im^j$ que minimize a função de erro E.

3. Um Resultado teórico

Um resultado interessante apresentado no teorema T-1 dá consistência teórica a metodologia proposta. Neste teorema mostra-se que qualquer função L^2-integravél pode ser aproximada por funções da forma $g^m(x)$. Dois resultados auxiliares, apresentados a seguir facilitam a prova deste resultado.

Resultado auxiliar RA-1: As funções simples, da forma $S(x) = \sum_{j=1}^{m} a_i X_{Ai}(x)$ com α_i constante real e $X_{Ai}(x)$ a função indicadora do conjunto Ai (isto é, toma-se o valor 1 se $x \in Ai$ e zero caso contrário) são densas em L^2 [3].

Resultado auxiliar RA-2: As funções $\{g^m(x)\}$ aproximam qualquer função simples. Isto é, $\forall$ S(x), existe uma seqüência de funções $g_n(x) \in \{g^m(x)\}$ tal que $g_n(x) \to S(x)$, onde esta convergência é referida à convergência em L^2

Prova do RA-2:

Seja $S(x) = X_{[0,\ 1]}$. Consideremos a seqüência onde $\kappa_n(x)=1$, isto é: $g_n(x) = B_n(x)$ com $h_1=0$ $h_2=1$.

$$B_n(x) = -C_n\left[\frac{1}{(1+e^{d_n x})} - \frac{1}{(1+e^{d_n (x-1)})}\right]$$

onde $C_n = \left(e^{d_n/2}+1\right)/\left(e^{d_n/2}-1\right)$ e $d_n \to \infty$ (para garantir que o máximo de $B_n(x)$ em [0 1] seja igual a 1)

1) É fácil provar que, $B_n(x) \xrightarrow[n]{} S(x)$ pontualmente, isto é $B_n(x) \to \begin{cases} 0 & x \notin [0,1] \\ 1 & x \in [0,1] \end{cases}$

Para chegar a convergência em L^2, partindo-se da convergência pontual, pode-se utilizar o teorema da convergência dominada de Lebesgue: (Se $f_n(x) \le g(x) \in L^2$, e $f_n(x) \to f(x)$ pontualmente então $f_n(x) \to f(x)$ em L^2) [3].

Como já se tem a convergência pontual, basta provar que $B_n(x)$ é dominada por uma função L^2-integrável.

É fácil verificar que a função L^2-integrável g(x) definida como:

$$g(x) = \begin{cases} 1+e^{x} & x < 0 \\ 2 & x \in [0,1] \\ 1+e^{-x} & x > 1 \end{cases}$$

limita a função $B_n(x)$ para todo x qualquer seja n, como se queria provar.

Resta ainda generalizar o intervalo [0 1] para um intervalo arbitrário, deve-se encontrar a seqüência Bn(x) que converge para funções do tipo X_A $A=[\delta_1, \delta_2]$. Para isso basta considerar as funções $B_n(x)$ como no caso anterior considerando $h_1=\delta_1$, $h_2=\delta_2$.

A extensão deste resultado para qualquer função simples S(x) segue da consideração de que a convergência em L^2 mantém-se com as operações de soma e multiplicação por uma constante. Como as funções simples são expressas como combinações lineares finitas de funções do tipo X_{Ai}, obteríamos uma seqüência $B_n(x)$ que converge a S(x) em L^2 considerando a combinação linear das funções $B^{(i)}{}_n(x)$ que convergem a X_{Ai}. ð

Note-se que na escolha de $g_n(x)$ só foram precisas funções $k_n(x)$ constantes. Por tanto, o teorema é válido inclusive nesse caso. A escolha da forma linear $\kappa(x) = ax + b$ (sub-índices omitidos) se dá unicamente para melhorar a aproximação e acelerar a convergência.

Teorema T-1:

Sejam $g^m(x) = \sum_{j=1}^{m} B_j(x) k_j(x)$ onde

$\kappa_j(x) = a_j x + b_j$,

$$B_j(x) \equiv -C_j \left[\frac{1}{1+\exp(d_j(x-h_j^{(1)}))} - \frac{1}{1+\exp(d_j(x-h_j^{(2)}))} \right]$$

j=1, . . . ,m.

Qualquer função L^2-integravél pode ser aproximada por funções da forma $g^m(x)$.

Prova:

O que se deseja provar é que o conjunto de funções $\{g^m(x)\}$ com norma $\|.\|_{L_2}$ é um conjunto denso em L^2. Deve-se provar que, para qualquer função em L^2, existe uma seqüência de funções $g_n(x) \in \{g^m(x)\}$ que converge a f(x) em L^2.

De **AR-1** tem-se que funções simples formam um conjunto denso em L^2. Assim Existe uma seqüência de funções simples, $S_k(x)$ que converge para f(x) em L^2.

De **AR-2**, tem-se que as funções simples, por sua vez podem ser aproximadas por funções do tipo $g^m(x)$. Para cada função simples $S_k(x)$, existe uma seqüência $g_{kn}(x)$ que converge para $S_k(x)$ em L^2:

$$\begin{array}{l} g_{11}(x)\ g_{12}(x) \ldots g_{1n}(x) \rightarrow S_1(x) \\ g_{21}(x)\ g_{22}(x) \ldots g_{2n}(x) \rightarrow S_2(x) \\ \quad : \quad : \qquad : \qquad : \\ g_{k1}(x)\ g_{k2}(x) \ldots g_{kn}(x) \rightarrow S_k(x) \end{array}$$

Para construir a $f_n(x)$ usa-se a diagonal de Cantor[4]: Escolhendo $g_n(x)=g_{nn}(x)$, então $g_n(x) \rightarrow f(x)$ em L^2. ð

Note-se que este Teorema foi provado utilizando a norma L^2 embora o problema foi exposto em termos da norma descrita em 2.2. Entretanto, para qualquer função g(x), definida pelos parâmetros $\Im$ =(C, d, $h^{(1)}$, $h^{(2)}$, a, b), a mesma função pode ser obtida através dos parâmetros $\Im^*$ onde $C^*_i=C_i/S$, $a^*_i=Sa_i$, $b^*_i=Sb_i$. $d^*=d$, $h^{*(1)}=h^{(1)}$, $h^{*(2)}=h^{(2)}$ e $S=\Sigma C_i$. Assim a nova função satisfará a condição $\Sigma C_i=1$ e portanto anula o segundo termo da norma 2.2. Desse modo, pode-se concluir que a mesma seqüência que aproxima as funções com a norma L^2, aproximará quando for considerada a norma 2.2.

4. Escolha inicial dos parâmetros

A relação entre a entrada e a saída da rede é aprendida através da estimação dos parâmetros que definem as funções de pertinência e de aproximação. As funções de pertinência podem se sobrepor em parte do domínio permitindo que um determinado ponto seja estimado através de uma combinação ponderada de mais de uma função de aproximação.

A escolha inicial dos parâmetros $h_1^{(1)}$ e $h_m^{(2)}$ pode refletir um conhecimento a priori do domínio da função. Pode-se ainda utilizar uma heurística de inicialização como a descrita abaixo com a finalidade de acelerar a convergência.

A idéia central da heurística que se segue é fazer uma divisão do domínio obtendo intervalos onde a função é aproximadamente monotônica. Para tal, ajusta-se, sobre os dados, um polinômio com grau igual ao número de funções de aproximação que se pretende utilizar. Calculando-se os máximos e mínimos do polinômio, determina-se as regiões do domínio onde a função mantém-se monotônica. Para definir os valores a e b associados à função de aproximação linear, para cada intervalo, ajusta-se, uma reta a través de uma regressão linear.

Um heurística automática para calcular a solução inicial:

Dados m, x_i, y_i

Passo 1: Ajustar aos dados o polinômio P(x) de grau m.

Passo2: Calcular os pontos máximos e mínimos de P(x): m_1, …m_m

Se $Im(m_i)=0$ $\forall i$, escolher $I_1=[\min(x_i)-\varepsilon, m_1-\varepsilon]$, $I_2=[m_1,m_2-\varepsilon]$,…, $I_m=[m_m, \max(x)+\varepsilon]$

Caso contrário, se algum m_i é complexo, dividir o intervalo em m intervalos uniformes disjuntos.

Passo 3: Para cada intervalo, ajustar regressão linear nos pares (x_i,y_i), $x_i \in I_i$ para obter a_i e b_i.

5. Resultados numéricos

O primeiro experimento numérico (veja figuras 3a; 3b; 3c; 3d) consiste de 100 pontos gerados pela função sen(x)+2 com diferentes níveis de ruído. O ruído foi gerado adicionando-se a função geradora um sinal Gaussiano de média zero e desvio padrão igual a 0.1; 0.4 e 0.7 em cada uma dos 3 experimentos. A tabela 5.1 resume os resultados destes experimentos. Em cada uma destas simulações foram utilizadas 3 pares de funções de pertinência - aproximação, parâmetros de tolerância. Em todos os experimentos, exceto no experimento 4 foi usada a heurística exposta na seção 4 para inicialização.

No	Nível de ruído	No. de épocas	Mape do ruído	Mape treinamento	Mape generalização
1	0	111	0	0.14	0.157
2	0.1	137	4.79	4.64	1.04
3	0.4	78	16.13	16.80	4.91
4	0.7	37	30.88	52.36	7.87

Tabela 5.1 Resultados das simulações com a função senx+2 com vários níveis de ruído.

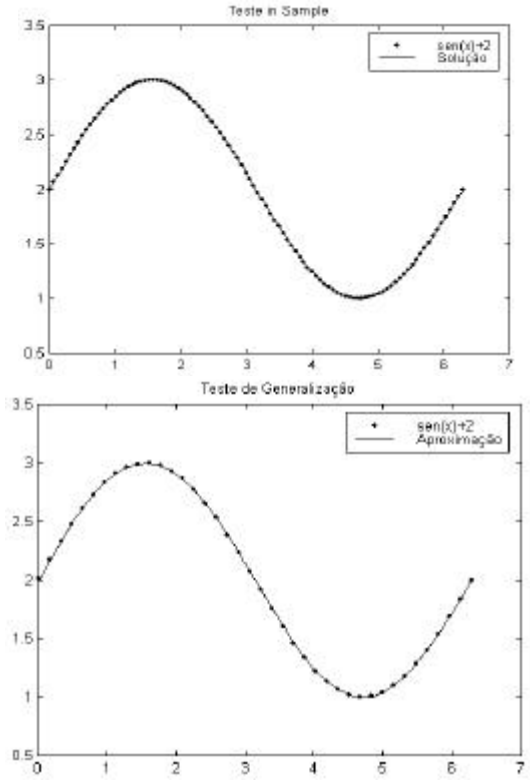


Figura 3a - Nível de ruído: 0

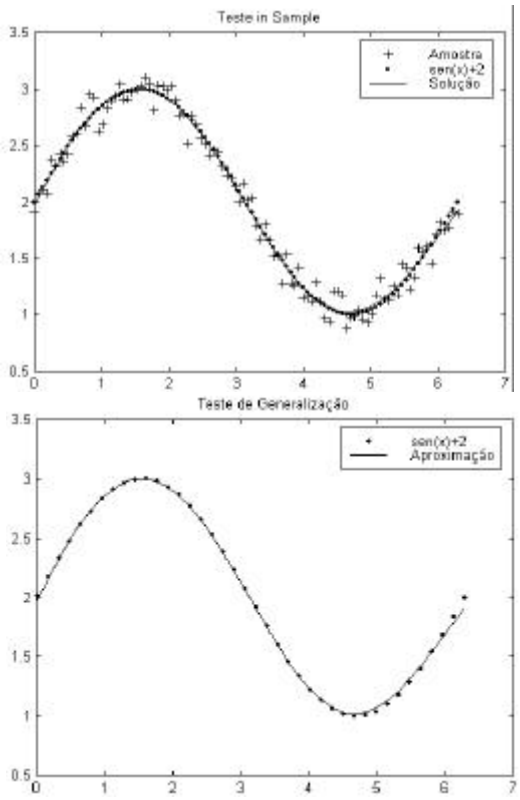


Figura 3b - Nível de ruído: 0.1

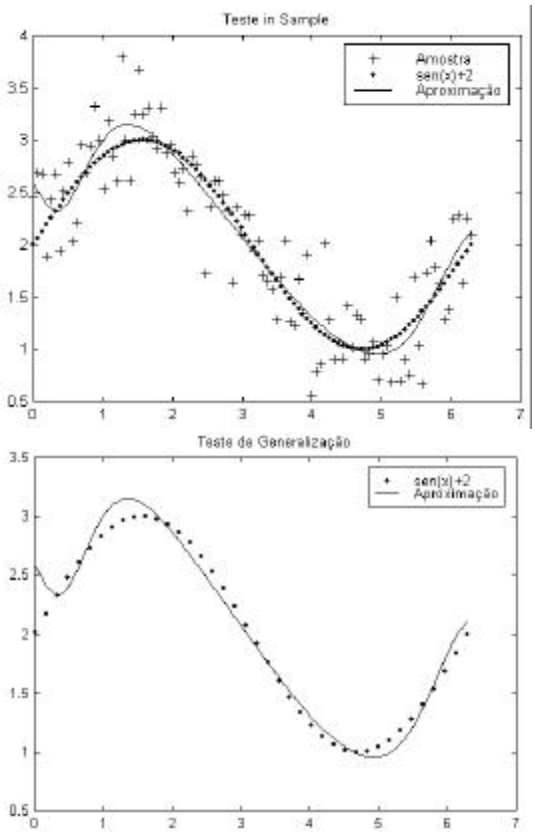


Figura 3c - Nível de ruído: 0.4

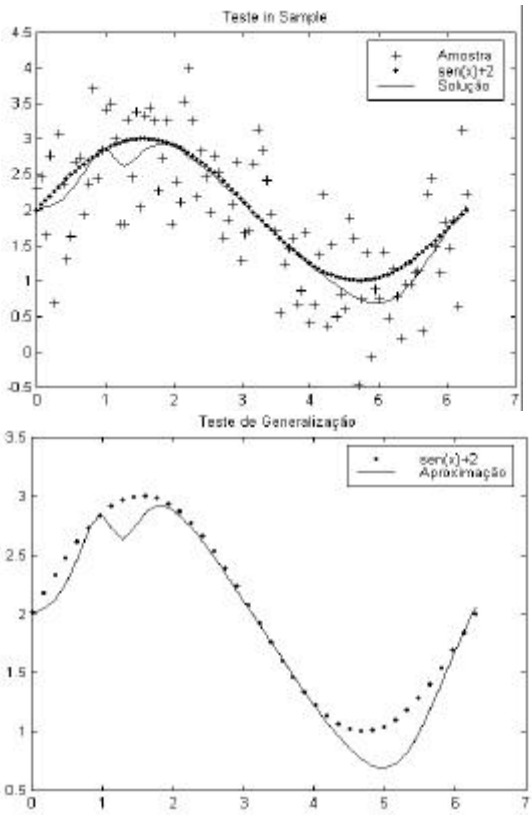
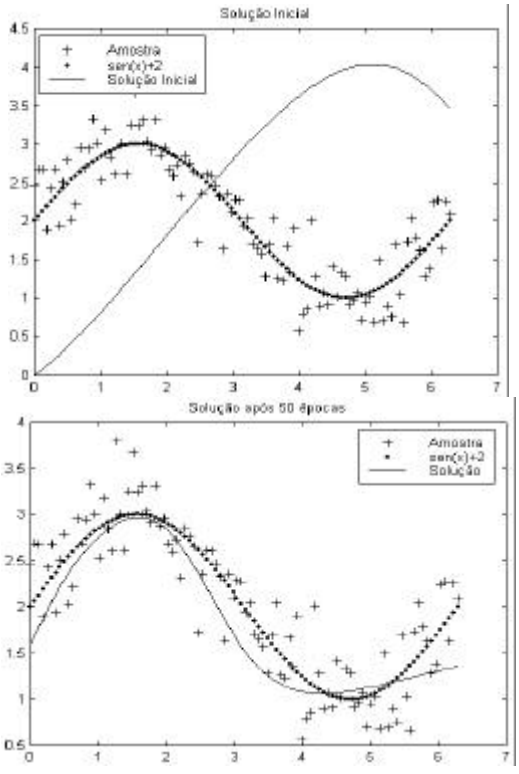


Figura 3d - Nível de ruído: 0.7

Utilizando os mesmos dados do experimento 4, tabela 5.1 inicializou-se o algoritmo, deliberadamente com uma condição inicial distante da solução esperada. Após 355 iterações obteve-se o Mape de 17.0na etapa de treinamento e 3.20 na etapa de generalização. A evolução da convergência neste caso é apresentada na figura 4.



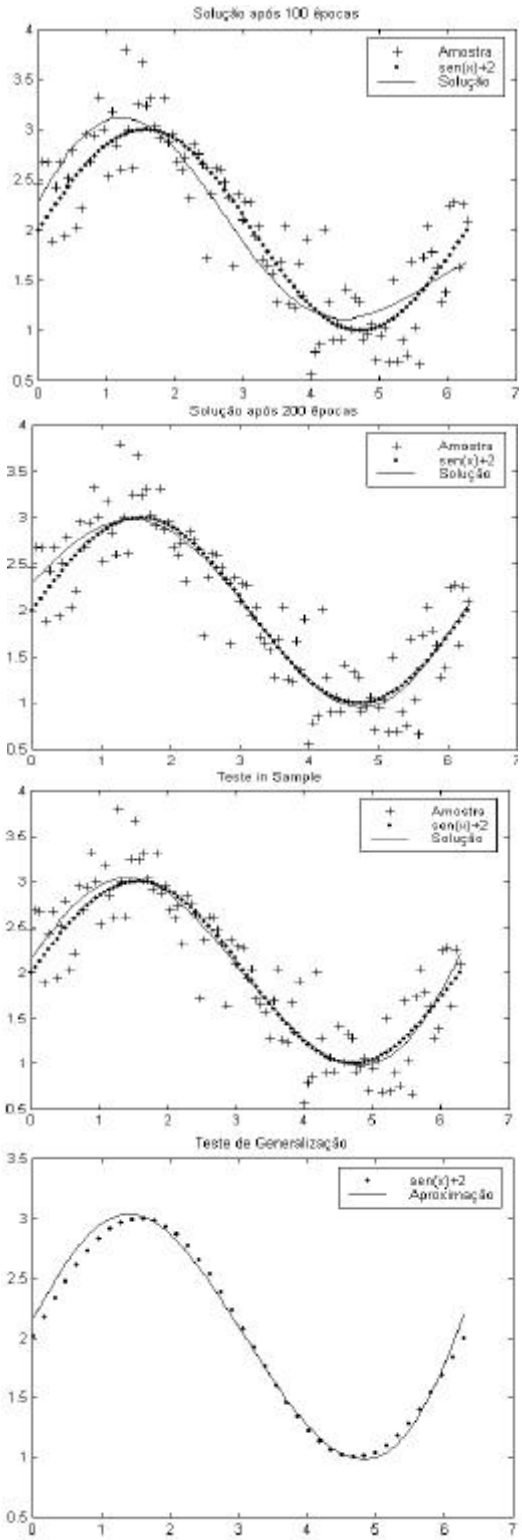


Figura 4 Evolução da convergência com inicialização sem uso da heurística

6. Uma aplicação em um problema de dados faltantes

Nesta seção apresenta-se uma aplicação da metodologia proposta em um problema de dados faltantes. Foram usados aqui dados reais de uma série de carga elétrica. A série considerada refere-se aos

dados de carga minuto a minuto para o dia 1 de Julho de 1999 da concessionária Light. Neste tipo de série o problema de dados faltantes é bastante comum.

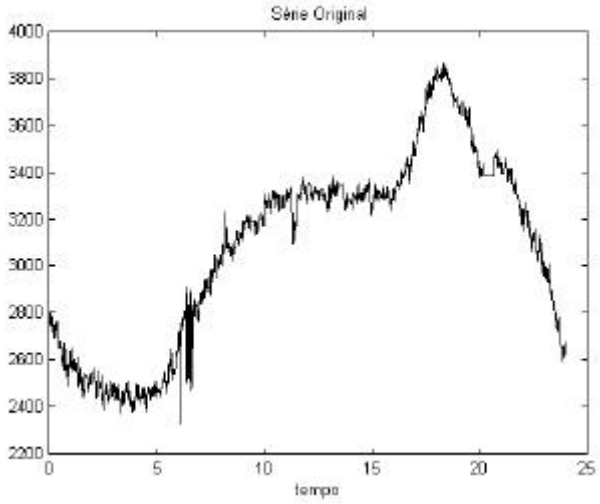


Figura 5 - Dados de carga minuto a minuto para o dia 1 de Julho de 1999

O experimento consiste em retirar da série original uma quantidade de pontos. A série é então recomposta pelo método proposto. Desta forma simula-se o problema de dados faltantes que é muito comum em várias aplicações e em particular no problema de previsão de carga. Foram retirados da série, aleatoriamente de modo uniforme, 5; 10; 20; 30 e 40 % de pontos em cada um dos 5 experimentos realizados. A tabela 6.1 resume o resultado encontrado.

Exp.	pontos retirados	Num. de épocas	Mape treinamento	Mape generalização
1	5%	75	0.99	1.21
2	10%	333	1.01	1.15
3	20%	106	1.06	1.116
4	30%	379	1.00	1.119
5	40%	105	1.07	1.117

Tabela 6.1 Simulação de dados faltantes com dados reais de carga elétrica

Em seguida fez-se uma comparação deste método com uma interpolação através de spline cúbico suavizado.

Nota-se (ver figura 7) que o spline não apresenta bons resultados quando se tem um número considerável de valores faltantes consecutivos. Este resultado era esperado uma vez que o spline fornece soluções com curvatura acentuada distanciando-se do padrão esperado de curva de carga diária.

Utilizou-se neste experimento comparativo o dia 1 de Julho omitindo-se 150 valores consecutivos. Após cerca de 400 épocas chegou-se ao resultado apresentado na figura 7. O MAPE obtido pelo método proposto foi de 1.02 na fase de treino e 2.19 no teste out of sample. Já a interpolação spline resultou em um MAPE de 9.31.

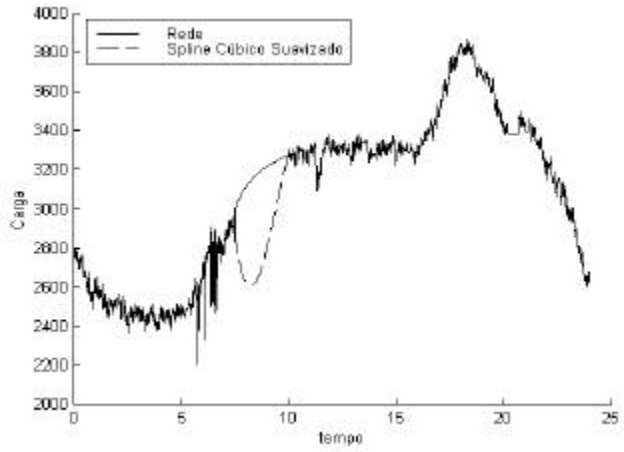


Figura 7 Comparação do método proposto com uma interpolação através de Spline Cúbico Suavizado

7. Conclusões

Neste artigo foi proposta uma arquitetura conexionista que possui uma interessante capacidade interpolativa. Sua principal característica é a capacidade intrínseca de gerar soluções regulares. Este tipo de arquitetura abre também a possibilidade de interpretabilidade dos resultados uma vez que a localização das funções de pertinência pode indicar uma mudança no modelo. Mudanças na estrutura da função geradora dos dados devem se refletir em mudanças de posicionamento e de nível das funções de pertinência.

Os resultados simulados em ambientes ruidosos foram particularmente bons. Embora em alguns experimentos uma quantidade razoável de ruído tenha sido adicionada, o método mostrou-se robusto produzindo ótimos resultados em especial na fase de generalização.

Do ponto de vista de aplicações reais o experimento para dados faltantes apresentou resultados muito animadores, mais uma vez demonstrando a capacidade do método de interpolar dados produzindo soluções regulares e consistentes.

Referências:

[1] Haykin S. “Neural Networks – A Comprehensive Foundation” , Prentice Hall, second edition, 1999.

[2] Pedroza L.C e Pedreira C.E. “Multilayer Neural Networks and Function Reconstruction by Using a priori Knowledge” International Journal of Neural Systems, Volume 9, number 3, pp 251-256, 1999.

[3] Bartle, R.G. Elements of integration. Wiley. New York, 1966.

[4] De Barra, G. (1974) Introduction to measure theory, New York, Van Nostrand Reinhold .