

REDES NEURAIS SEGMENTADAS – UMA APLICAÇÃO AO PROBLEMA DE RECONHECIMENTO DE PADRÃO

Luiz Carlos Pedroza

CEFET-RJ Av. maracanã, 229 Rio de Janeiro, CEP 20271-110

E-mails, pedroza@cefet-rj.br

Resumo— Reconhecimento de padrão é um problema de grande relevância na área de processamento de sinais. Geralmente o padrão procurado está deformado e embutido em um outro sinal. Neste artigo é apresentada uma solução para este problema a partir de uma nova arquitetura de rede neural que sendo treinada a priori com um padrão modelo, permite sua deformação sistemática a medida que opera a busca pelas suas deformações embutidas no sinal disponível.

Abstract— Pattern recognition is a very relevant problem in signal processing area. Generally the searched pattern is distorted and built into another signal. In this paper it is presented a solution to this problem through the use of a new neural network architecture which, trained a priori with the model pattern, allows its systematic distortions while accomplishing its search into the available signal

Keywords—Neural Networks, Pattern Recognition

1 Introdução

Um problema de grande relevância na área de processamento de sinais é o problema de reconhecimento de padrão já que o padrão procurado está geralmente deformado e embutido em outro sinal (Lai and Chin,1995;Lai and Chin,1998). Geralmente este problema é abordado, na área de redes neurais, através do uso de redes recursivas (Lippmann,1989a;Lippmann,1989b). No caso, o padrão a ser reconhecido define um atrator na função de energia da rede recursiva. Contudo pela dinâmica não linear da rede o problema de geração destes campos atratores é extremamente complexo fazendo com que geralmente, nos casos práticos, o sinal disponível, onde se opera o reconhecimento, não possa diferir muito do padrão a ser reconhecido(Haykins,1999)

Como mostrado em (Pedroza, Pedreira e Fariñas,2001), as Redes Neurais Segmentadas (RNS) representam uma nova arquitetura conexionista com performance, no caso de aproximação de funções, equivalente a das redes multicamadas tradicionais. Sua arquitetura foi idealizada para permitir o particionamento do domínio da função a ser emulada enquanto dentro destes particionamentos a função é aproximada por um conjunto de funções de apoio muito simples, muitas vezes lineares. Em outras palavras, a idéia central da RNS é expressar o mapeamento entrada-saída desejado através de uma função segmentada em intervalos. Sua estrutura básica é constituída pela combinação de vários neurônios compostos por uma funções de aproximação e uma 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.

Devido a sua propriedade de particionamento do domínio de aproximação, devido a possibilidade de análise da pertinência de cada neurônios em um ponto do domínio e pela simplicidade das funções de ativação as redes segmentadas permitem uma melhor extração do conhecimento contido na sua estrutura do que as tradicionais(Pedreira, Pedroza e Fariñas, 2001). Esta propriedade é de grande valia quando da sua aplicação ao problema de reconhecimento de padrões pois permite que se proponha transformações na estrutura da rede, previamente treinada com um padrão modelo, de forma que esta possa ser re-treinada, usando-se agora um conjunto de treinamento obtido do sinal real disponível. Este re-treinamento realizará uma busca, sobre sinal real, de deformações do padrão modelo.

2 A arquitetura Proposta

Consideremos uma rede com m nós ou neurônios. Seja $\{x_i\}_i^p$ a partição dos dados usada para treinamento. Por simplicidade algébrica e de notação iremos considerar o caso onde $x \in \mathfrak{X}$ (o subscrito de x será omitido), a generalização para o caso onde

$x \in \mathcal{R}^n$ é algebricamente direta. Define-se, para cada ponto x , m funções de pertinência do seguinte modo:

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

$j=1..,m$

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 a 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).

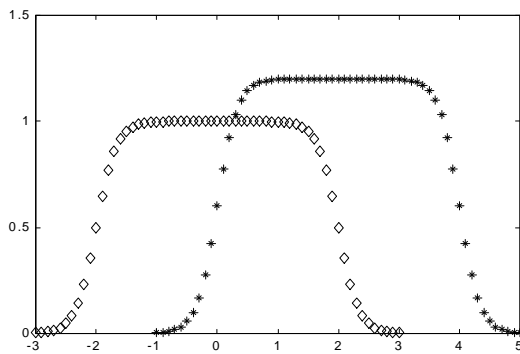


Figura 1 – Exemplos de funções de ativação

◇ $C=1$, $d=6$; $h_j^{(1)} = -2$; $h_j^{(2)} = 2$

* $C=1.2$, $d=6$; $h_j^{(1)} = 0$; $h_j^{(2)} = 4$

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:

$$\kappa_j(x) = a_j x + b_j \quad j=1, \dots, m$$

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 se 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 efetuada 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:

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

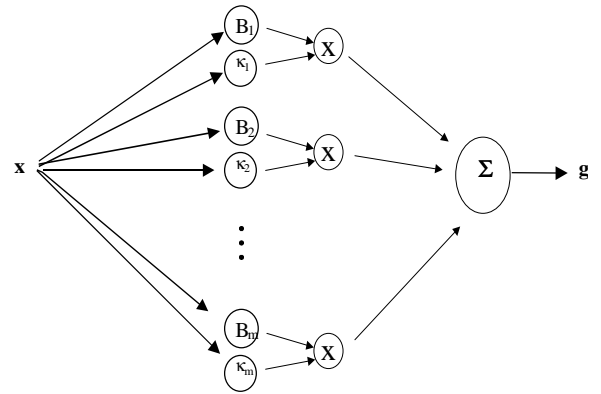


Figura 2 – A arquitetura proposta

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

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 :

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

onde

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

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 $\mathcal{J}^j \equiv (C_j, d_j, h_j^{(1)}, h_j^{(2)}, a_j, b_j)$ e o objetivo central será encontrar os \mathcal{J}^j ótimos que minimizem a função de erro E . Isto pode ser realizado utilizando-se do gradiente decrescente ou outro algoritmo análogo.

Em Pedroza *et alli* (2001) prova-se que o conjunto de funções gerados pela RNS é denso em L_2 . Ou seja a rede se presta plenamente a aproximação de funções quadraticamente integráveis (Naylor and Sell, 1982).

3. Escolha inicial dos parâmetros e aceleração do treinamento.

A relação entre a entrada e a saída da rede é aprendida através da variação dos parâmetros que definem as funções de pertinência e de aproximação. As funções de pertinência podem se sobrepor em parte do domí-

nio 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 com 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.

3.1 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, \dots, m_m

Se $\text{Im}(m_i)=0 \forall i$, escolher $I_1=[\min(x_i)-\epsilon, m_1-\epsilon]$, $I_2=[m_1, m_2-\epsilon], \dots, I_m=[m_m, \max(x)+\epsilon]$

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 .

Ao se utilizar este procedimento para inicialização dos parâmetros da rede, as regiões de pertinência dos neurônios ficam suficientemente disjuntas. Isto faz com que apenas alguns neurônios da vizinhança de um determinado ponto x do domínio da função $f(x)$ possuam função de pertinência considerável. Neste caso, todos os outros neurônios que não estejam nesta vizinhança apresentam saída praticamente nula e podem ser desconsiderados durante esta interação em x do algoritmo de treinamento.

3.2 Exemplo

O exemplo em questão consiste em uma conjunto de treinamento de 100 pares obtidos da função $f(x)=\text{sen}(x)+2$ corroida por um ruído Gaussiano de media zero e desvio padrão .1 e definida sobre o intervalo $0 < x < 2\pi$. A configuração inicial da rede é gerada pela heurística proposta. A figura 3a mostra o aproximação gerada pela rede. Na figura 3b mostra-se que mesmo na presença de ruído a RNS apresenta uma boa capacidade de generalização.

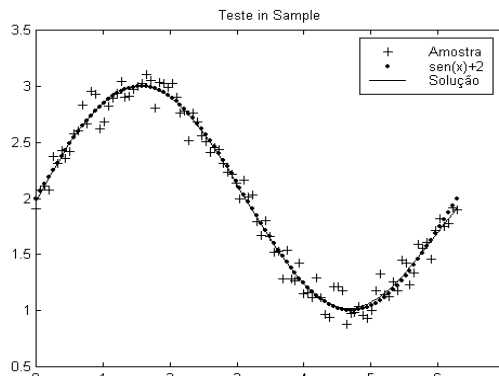


Figura 3a -

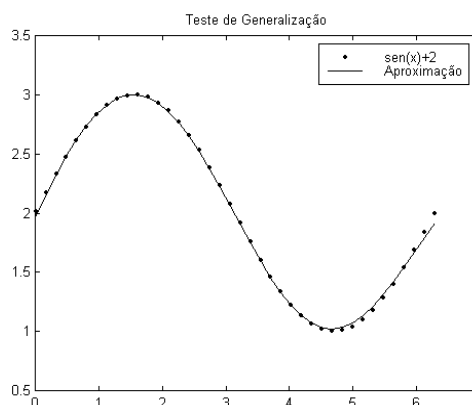


Figura 3b - Nível de Ruído: 0.1

4. Reconhecimento de Padrão

Observe no exemplo da seção 3.1 que, partindo-se da RNS que realize a função $\text{sen}(x)+2$ no intervalo $0 < x < 2\pi$ é possível operar transformações nesta rede de forma que ela aproxime a função $3\text{sen}(2x+4)+5$ no intervalo $-2 < x < \pi-2$, para isto basta se alterar, pela introdução e modificação de sinapses, as funções de aproximação para $3\kappa_j(2x+4)$ e as de pertinência para $B_j(2x+4)$ e somar 3 ao neurônio de saída da rede. Esta possibilidade de alteração sistemática dos pesos da rede permite se gerar a aproximação de toda uma família de funções. Esta propriedade, quando aplicada ao problema de reconhecimento de padrão, irá permitir a deformação sistemática de um padrão modelo de forma que toda uma família do padrão, que na verdade são suas deformações, possam ser descrita pela rede alterada. Em seguida a rede será retreinada de forma a buscar dentro desta família o padrão deformado que possa estar embutido dentro do sinal disponível para análise.

Seja $M: \mathfrak{X}^n \rightarrow \mathfrak{X}^m$ o mapeamento do padrão modelo $M(x)$ definido sobre o intervalo $I_1 \leq x \leq I_2$.

Seja

$$P(x) = \beta_0 M(\sigma_0 x + \tau_0) + \lambda_0 \quad (4.1)$$

uma família de padrões deformados definidos sobre o intervalo

$$l_1 \leq (\sigma_0 x + \tau_0) \leq l_2 \quad (4.2)$$

Seja $s(x)$ o sinal, que contem o padrão deformado, definido em \mathfrak{R}^n como:

$$\begin{cases} s(x) = \beta_1 M(\sigma_1 x + \tau_1) + \lambda_1 & l_1 \leq (\sigma_1 x + \tau_1) \leq l_2 \\ s(x) \neq \beta_2 M(\sigma_2 x + \tau_2) + \lambda_2 & \text{caso contrario} \end{cases} \quad (4.3)$$

Nas definições

$$\beta_i \in \mathfrak{R}^m \times \mathfrak{R}^m, \sigma_i \in \mathfrak{R}^m \times \mathfrak{R}^n, \tau_i \in \mathfrak{R}^m, \lambda_i \in \mathfrak{R}^m \quad \forall i = \{0,1,2\}$$

são operadores lineares.

O problema de reconhecimento de padrões é aqui definido como a busca de valores ótimos $\beta_0^*, \sigma_0^*, \tau_0^*, \lambda_0^*$ que minimizem a norma

$$\|P(x) - s(x)\| \quad (4.4)$$

sobre o intervalo

$$\frac{l_1}{\sigma_0} - \tau_0 \leq x \leq \frac{l_2}{\sigma_0} - \tau_0. \quad (4.5)$$

Este problema de minimização pode ser resolvido por métodos de otimização a partir da escolha dos pontos x dentro da região de viabilidade dada por (4.5)

Observa-se que se

$$\frac{l_1}{\sigma_0} - \tau_0 < \frac{l_1}{\sigma_1} - \tau_1 < \frac{l_2}{\sigma_1} - \tau_1 < \frac{l_2}{\sigma_0} - \tau_0 \quad (4.6)$$

o problema apresenta apenas uma solução global

$$\beta_0^* = \beta_1, \sigma_0^* = \sigma_1, \tau_0^* = \tau_1, \lambda_0^* = \lambda_1$$

Se a condição (4.6) não for satisfeita o problema poderá convergir para outras soluções com por exemplo para $\beta_0 = 0$ ou $\sigma_0 = \infty$

Para se aplicar as RNS na solução do problema de reconhecimento de padrões, o primeiro passo consiste em se treinar a rede com o padrão que se deseja reconhecer: a função M . Após o treinamento gerar uma aproximação satisfatória o segundo passo consiste em se alterar a rede já treinada de forma que seu novos parâmetros de treinamento sejam $\mathfrak{S}_0 = (\beta_0, \sigma_0, \tau_0, \lambda_0)$. Isto se faz alterando-se as função de aproximação para $\beta_0 \kappa_j(\sigma_0 x + \tau_0)$ e as de pertinência para $B_j(\sigma_0 x + \tau_0)$ e se adicionando o peso sináptico λ_0 ao neurônio da camada de saída da rede. Neste caso a equação 2.1 é reescrita como:

$$P(x) = \left(\sum_{j=1}^m B_j(\sigma_0 x + \tau_0) \beta_0 \kappa_j(\sigma_0 x + \tau_0) \right) + \lambda_0 \quad (4.7)$$

O vetor de parâmetros inicial \mathfrak{S}_0 é escolhido de forma a satisfazer a condição 4.6.

A próxima etapa consiste em sortear pontos x que satisfaçam 4.5 e utilizando-se de um método de otimização variar os pesos $\beta_0, \sigma_0, \tau_0, \lambda_0$ a fim de minimizar a norma 4.4.

5. Exemplos

Nesta seção, apresentam-se dois exemplos do uso da metodologia proposta a problemas em \mathfrak{R}^1 de reconhecimento de padrões. O primeiro exemplo, mais simples, trata de reconhecimento de um padrão estático embutido em um sinal ruidoso. O segundo mostra a capacidade de se utilizar desta metodologia para se rastrear um padrão não-estático sobre um sinal mais complexo.

Exemplo 5.1

Seja o padrão modelo

$$M(x) = \begin{cases} 0.95, & 0 \leq x \leq .5 \\ 0.7, & .5 < x \leq 1 \\ 1.9, & 1 < x \leq 1.5 \\ 1.2, & 1.5 < x \leq 2 \end{cases}$$

Deseja-se reconhecer este padrão embutido e deformado no sinal

$$s(x) = \begin{cases} 0.45, & 1 \leq x \leq 1.25 \\ 0.35, & 1.25 < x \leq 1.5 \\ 0.9, & 1.5 < x \leq 1.75 \\ 0.6, & 1.75 < x \leq 2 \\ 0.1\epsilon & \text{caso contrario} \end{cases}$$

onde ϵ é um ruído branco com variância unitária.

O primeiro passo consiste em se treinar, a partir do que foi discutido nas seções 2 e 3, uma rede segmentada para aproximar o padrão modelo M . A figura 4 mostra o padrão modelo aproximado pela RNS.

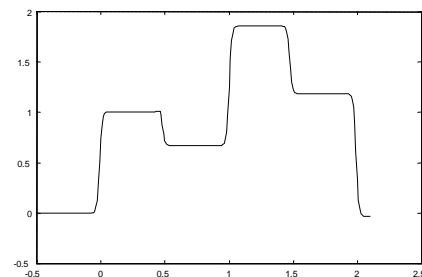


Figura4 -Padrão modelo M aproximado pela rede

O segundo passo consiste em se alterar as funções de ativação e pertinência da rede de forma que esta seja capaz de realizar toda a família de padrões deformados

$$P(x) = \beta_0 M(\sigma_0 x + \tau_0) + \lambda_0.$$

O terceiro passo consiste na escolha de valores iniciais para os novos parâmetros da rede

$$\beta_0 = 1, \sigma_0 = 1, \tau_0 = 0, \lambda_0 = 0$$

Para demonstrar a robustez do processo de reconhecimento de padrão a escolha dos parâmetros iniciais da rede alterada não satisfaz a condição 4.6, como pode se ver na figura 5 que mostra um instante do processo de busca em uma de suas interações iniciais.

Por ultimo, aplica-se o algoritmo de treinamento para buscar os parâmetros $\mathfrak{S}_0 = (\beta_0, \sigma_0, \tau_0, \lambda_0)$ ótimos da rede alterada. No caso, para acelerar o processo, apenas os neurônios na vizinhança de ponto x sorteado em uma determinada interação do algoritmo são levados em consideração para calcular as variações nos parâmetros $\beta_0, \sigma_0, \tau_0, \lambda_0$.

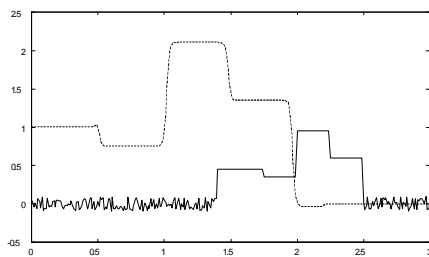


Figura 5- Interação inicial do processo de busca de $P(x)$ em $s(x)$

A figura 6 mostra a convergência do algoritmo de busca depois de 10k interações com o efetivo reconhecimento do padrão desejado. Observa-se que, mesmo com deformações no padrão e com a não satisfação da condição 4.6 o algoritmo de busca foi robusto o suficiente para reconhecer o padrão embutido em $s(x)$.

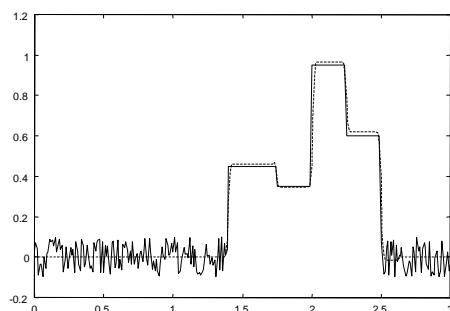


Figura 6 - final do treinamento $P(x)$ em $s(x)$

Exemplo 5.2

Suponha que agora se deseje aplicar esta técnica não ao reconhecimento de um padrão estático embutido em um sinal ruidoso, mas ao problema do reconhecimento e rastreamento de um padrão que se desloca continuamente sobre um sinal. No caso suponha que se deseje buscar o mesmo padrão do exemplo anterior embutido no sinal $s(x,t)$ definido como.

$$s(x,t) = \begin{cases} 0.45, & 1 \leq x - \delta t \leq 1.25 \\ 0.35, & 1.25 < x - \delta t \leq 1.5 \\ 0.9, & 1.5 < x - \delta t \leq 1.75 \\ 0.6, & 1.75 < x - \delta t \leq 2 \\ \frac{\text{signal}(1 + \sin(20\pi))}{2} & \text{caso contrario} \end{cases}$$

onde a função signal vale 1 se seu argumento for maior ou igual a zero e -1 caso contrario.

Note que o sinal $s(x,t)$ é variante no tempo. No caso, o padrão embutido se desloca com o tempo para esquerda com velocidade δ sobre uma seqüência de pulsos com largura $10/\pi$.

Usando o mesmo algoritmo de busca do exemplo anterior com condições iniciais idênticas o processo de reconhecimento e rastreamento é efetivado. Na figura 7 mostra-se uma fase onde o reconhecimento do padrão ainda não é satisfatório. No caso o padrão deformado encobre dois pulso da figura de fundo. Na figura 8, depois de 50k interações o padrão já foi reconhecido e o processo de rastreamento está em curso. No caso o padrão encobre apenas parcialmente dois pulsos.

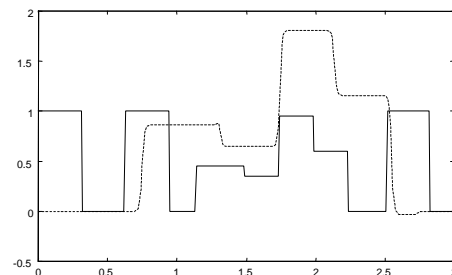


Figura 7- Interação inicial do processo de busca de $P(x)$ em $s(x)$

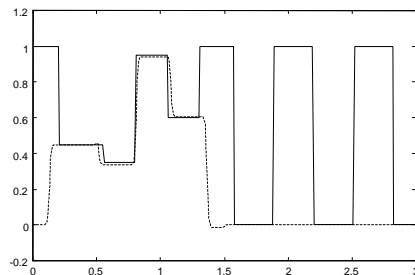


Figura 8 - rastreamento $P(x)$ em $s(x,t)$

Pelo exemplo pode-se perceber que o padrão procurado pode ser deslocado de forma continua sobre um sinal complexo que ainda assim, a partir

do processo de reconhecimento, é possível se realizar um efetivo rastreamento do padrão.

6. Conclusões

Neste artigo, baseando-se em uma nova arquitetura de rede neural, se propôs uma metodologia para efetuar o reconhecimento de padrões embutidos em sinais. No exemplo 5.1 mostra-se como treinar uma RNS para aprender um padrão modelo. Em seguida esta informação contida na rede é utilizada para identificar este padrão deformado e embutido em um meio ruidoso. No exemplo 5.2 mostra-se uma aplicação diferente onde o mesmo padrão deformado e deslocado para esquerda com velocidade constante e o processo de reconhecimento é capaz de acompanhar este deslocamento.

Ainda que os exemplos utilizados revelem uma baixa complexibilidade pertencendo a \mathfrak{R}^1 , os resultados alcançados são bastante promissores já que, como foi comentado no artigo, caso um padrão mais complexo demandasse mais neurônios para ser aproximado este acréscimo de neurônios não acarretaria um aumento exponencial no tempo de treinamento da rede pois a atualização dos pesos da rede se basearia apenas em neurônios na vizinhança do ponto x de treinamento.

Nas aplicações de processamentos de imagem onde comumente muitos padrões estão embutidos no mesmo quadro esta técnica parece bem promissora pois nada impede a busca paralela de vários padrões diferentes por varias redes diferentes sobre o mesmo conjunto de dados. A técnica parece bem promissora nas aplicações de controle via imagem pois nesta existem poucos padrões a serem reconhecidos e a capacidade de rastrear padrões moveis sobre fundos de imagem complexos seria bem útil na medida das variáveis de controle.

Quando comparada as técnicas tradicionais de reconhecimento de padrão via redes recursiva esta metodologia apresenta a deficiência de que para cada padrão a ser procurado é necessária uma rede pre-treinada, o que a principio acarretaria um acréscimo sistemático no número de neurônios no sistema. Entretanto como estas redes podem ser trabalhadas independentemente, até em computadores diferentes, esta dificuldade seria compensada pela maior capacidade de reconhecimento da metodologia proposta.

Os próximos passos da pesquisa consistem em: primeiro ampliar a aplicação da metodologia a problemas em \mathfrak{R}^2 ; descobrir metodologias para particionar ao máximo o problema permitindo a descentralização do uso dos recursos computacionais; investigar a utilização desta técnica na restaurações de padrões deformados por filtragem ou perda de parte de informação.

Referências

Haykin S. (1999). *Neural Networks – A Comprehensive Foundation*, 2nd ed. Prentice Hall New York

Lai K.F., Chin R.T.(1995), Deformable Contours: Modelling and Extraction ,*IEEE trans. Pat. Anal. Mach. Intell.* Vol-17,pp1084-1090

Lai K.F, Chin R.T.(1998), On Modelling, Extraction, Detection and Classification of deformable Contours from Noise Images, *Image and Vision Computing* Vol- 16 ,pp 55-62

Lippmann R.P(1989a), Pattern Classification using Neuron Networks *IEEE Communication Magazine*, Vol 27, pp 47-64, 1989

Lippmann R.P(1989b),Review of Neural Networks for speech recognition, *Neural Computation*, Vol-1,pp 1-38

Naylor A.W., Sell G.R.(1982), *Linear Operator Theory in engineering and science*. 2nd ed. Springer-Verlag, New York, 1982.

Pedreira C.E., Pedroza L.C., Fariñas M(2001a). Local-Global Neural Networks for Interpolation, *Proceedings of V International Conference on Neural Networks and Genetic Algorithms*, Spring Velag , Praga, Republica Theca, Vol 1, pp 230-235.

Pedroza L.C e Pedreira C.E(1999). Multilayer Neural Networks and Function Reconstruction by Using a priori Knowledge *International Journal of Neural Systems*, Vol- 9, number 3, pp 251-256.

Pedroza L C. e Pedreira C.E, Farinas M.(2001b) Redes Neurais Locais Globais-Uma alicação ao problema de dados faltantes, *Anais do V Congresso Brasileiro de Redes Neurais*, Rio de Janeiro Vol 1, pp 433-338.