

ISBRN
ISBRN
ISBRN

II S B R N

II Simpósio Brasileiro de Redes Neurais



São Carlos

18 a 20 de outubro de 1995

Modelos de Redes Neurais - II

Investigação de Propriedades Fundamentais em Redes RAM Radial	187
<i>Anne Magály de P. Canuto, Edson C. B. Carvalho Filho e Juliana N. Gouvêia Ribeiro (UFPE)</i>	
Variações de um Modelo de Rede Neural Competitiva de Hopfield para Aproximação Poligonal	193
<i>Aluizio F.R. Araújo (USP-SC) e Júlia Sawaki Tanaka (UNESP-Araraquara)</i>	
Métodos de rotulagem para as Redes de Kohonen	199
<i>Luiz Carlos da Silva e Edson C. de B. Carvalho Filho (UFPE)</i>	
Medida de Confiabilidade para Adaptive Logic Networks	205
<i>Sérgio A. de Figueiredo e Weber Martins (UFG)</i>	

Modelos de Redes Neurais - III

Computational Complexity and Design of Artificial Neural Networks	213
<i>M.C.P. de Souto e W.R. de Oliveira (UFPE)</i>	
Redes Neurais e Neurônios Nebulosos: Modelos e Aprendizagem	219
<i>M. Figueiredo, F. Gonide (UNICAMP) e W. Pedrycz (University of Manitoba)</i>	
Profiles of Neural Activity in Topographical Maps of Orientation Specificity	225
<i>Luciano da Fontoura Costa e Lidério Ioriatti (USP-SC)</i>	
A Rede Neural dos Vizinhos mais Próximos	231
<i>Emmanuel Passos e Romildo Valente (IME-RJ)</i>	

Redes Neurais em Reconhecimento de Padrões - I

VARIAÇÕES DE UM MODELO DE REDE NEURAL COMPETITIVA DE HOPFIELD PARA APROXIMAÇÃO POLIGONAL

Aluizio F. R. Araújo
Departamento de Engenharia Elétrica - USP
Av. Dr. Carlos Botelho, 1465
CEP 13.560-250 - São Carlos - SP
e-mail: aluizioa@peterpan.sel.eesc.sc.usp.br

Júlia Sawaki Tanaka
Instituto de Química - UNESP
Rua Prof. Francisco Degni, s/n
CEP 14.800-900 - Araraquara - SP
e-mail: julia@ifqsc.sc.usp.br

Abstract: This paper aims at modifying a model called Competitive Hopfield Neural Network used to polygonal approximation. Such a model showed performance limitations on polygons sharing particular features. The authors of this paper introduced two different strategies to construct the matrix of connections and use a supervised learning method for undesirable responses. Simulations illustrated improvements in the model performance.

Resumo: Este artigo enfoca aproximação poligonal para utilização em reconhecimento de padrões e visão computacional. Foram propostas algumas modificações na Rede Competitiva de Hopfield para aproximação poligonal visando diminuir as limitações do algoritmo original. Primeiramente, mudou-se a maneira de montar a matriz de conexões. Em seguida, introduziu-se treinamento supervisionado para casos em que a solução encontrada não é a desejada. Em simulações, observou-se melhores resultados em casos onde o algoritmo original é limitado.

1. Introdução

A representação de formas de objetos é crucial em reconhecimento de padrões e visão computacional. O emprego de formas aproximadas de objetos permite sua descrição de forma mais compacta reduzindo demanda de memória computacional, simplificando sua análise, reduzindo a quantidade de dados processáveis e preservando informações relevantes.

A aproximação poligonal é uma abordagem bastante utilizada para aproximação de formas de objetos no espaço bidimensional. Visa-se esboçar um polígono cujos vértices sejam alguns dos pontos dados de uma curva original.

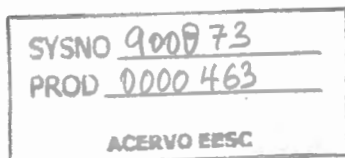
A dificuldade em resolver este tipo de problema consiste na escolha dos vértices do polígono de aproximação, de maneira que se tenha a melhor aproximação da curva. Este é um problema que tem várias abordagens diferentes, e

em geral requerem muita memória computacional e processamento complexo. Ainda assim, muitas vezes resulta em soluções não satisfatórias.

Modelos de redes neurais são empregados em aproximação poligonal onde substituem processos de otimização.

Este artigo propõe variações de um modelo de Rede Competitiva de Hopfield para aproximação poligonal (Chung e outros, 1994). Tal modelo mostrou limitações quando o número de vértices do polígono de aproximação (m) é pequeno, quando m é menor do que o número de vértices mais importantes para construção do polígono e/ou quando o objeto não é convexo.

As mudanças propostas envolvem modificações na matriz de conexões (H) da rede neural. Propõe-se, duas maneiras diferentes de construir H de modo que as diferenças entre o polígono de aproximação e a curva original sejam mais precisamente consideradas. Utiliza-se



1 900873
030696

também um algoritmo supervisionado para se obter uma aproximação particular da curva a partir de uma aproximação inicialmente resultante.

A próxima seção explica o modelo Rede Neural Competitiva de Hopfield e apresenta algumas de suas limitações. Na seção 3, propõe-se as modificações para a matriz de conexões do sistema. A seção 4 mostra resultados ilustrativos de simulações comparativas das abordagens distintas. A última seção discute os resultados encontrados.

2. Uso de Rede Neural Competitiva de Hopfield em Aproximação Poligonal

Esta seção apresenta um método de aproximação poligonal usando Rede Competitiva de Hopfield para achar os vértices do polígono.

Esta rede tem arquitetura bidimensional de n linhas e m colunas. Apenas uma unidade por coluna fica ativa ao final do processo. Cada linha pode ter no máximo uma unidade ativa. As unidades ativas representam os pontos da curva selecionados como vértices do polígono de aproximação. O algoritmo de Hopfield (1982) assegura que no máximo uma unidade por linha esteja ativa, enquanto que o algoritmo competitivo (Rumelhart e Zisper, 1985) garante que unidades diferentes de uma mesma coluna não sejam associados a um mesmo vértice do polígono de aproximação. A convergência desta rede é provada pelos autores.

Dada uma curva fechada $P = \{p_1, p_2, \dots, p_n\}$ consistindo de n pontos ordenados, no sentido horário. Nesta curva, $p_x p_y$ e $\overline{p_x p_y}$ representam respectivamente o trecho de curva e a corda do ponto p_x ao ponto p_y , no sentido horário. A desvio de um ponto dado (p) da curva à corda (desvio ponto-corda) denotado por $h_{x,y}$ é definido como:

$$h_{x,y} = \max_{p \in p_x p_y} \{d(p, \overline{p_x p_y})\} \quad (1)$$

onde: $d(p, \overline{p_x p_y})$ é a distância perpendicular do ponto p do trecho da curva $p_x p_y$ até a corda $\overline{p_x p_y}$.

Defin-se então H como sendo a matriz formada por tais desvios:

$$H = \{h_{x,y}\}, \quad x, y = 1, 2, \dots, n$$

Neste caso, o desvio ponto-corda é utilizado para avaliar a proximidade entre o polígono de aproximação e a curva.

Dadas a curva e um número m , o método visa encontrar o polígono de aproximação $Q^{(m)}$ com m vértices dentre os pontos dados da curva, de forma que a soma de todos os desvios ponto-corda seja mínima.

Na rede neural as linhas representam os n pontos da curva e as colunas os m vértices do polígono de aproximação. O valor $V_{x,i}$ representa o estado binário de ativação da unidade (x,i) , onde $V_{x,i} = 1$ indica que o ponto p_x é o i -ésimo vértice do polígono de aproximação. Nesta rede, a matriz H descreve a influência de uma unidade sobre a mesma e bidirecionalmente sobre todas as outras na sua própria coluna.

Aplicando a regra "winner-take-all", as unidades de uma coluna competem com as demais na mesma coluna para ativar aquela que recebe o maior valor de soma ponderada. A vencedora fica com estado 1, e as demais ficam com estado 0. Esta regra garante que dois pontos diferentes p_x e p_y da curva não são associados ao mesmo vértice do polígono de aproximação, e que exatamente m vértices são selecionados.

Por tanto, a regra de ativação para a i -ésima coluna é dada por:

$$V_{x,i} = \begin{cases} 1 & \text{se } U_{x,i} = \max\{U_{1,i}, \dots, U_{n,i}\}, \\ 0 & \text{se } U_{x,i} \neq \max\{U_{1,i}, \dots, U_{n,i}\}. \end{cases} \quad (2)$$

onde $U_{x,i}$ é a soma ponderada da unidade (x,i) :

$$U_{x,i} = \sum_{j=1}^m -(h_{y,x} V_{y,j-1} + h_{x,y} V_{y,i+1}) \quad (3)$$

onde os índices são cíclicos, de tal forma que $V_{0,i} = V_{n,i}$; $V_{n+1,i} = V_{1,i}$; $V_{x,0} = V_{x,m}$ e $V_{x,m+1} = V_{x,1}$.

No modelo, as unidades de uma determinada coluna só recebem entradas da coluna anterior e da coluna posterior, o que implica na redução significativa da complexidade da rede.

Visando-se que nenhum ponto p_x ocorra em mais que uma posição i no polígono de aproximação, o desvio ponto-corda é modificado para:

$$h_{x,y} = \begin{cases} \infty & \text{se } x = y, \\ \max_{p \in p_x p_y} \{d(p, \overline{p_x p_y})\} & \text{se } x \neq y. \end{cases} \quad (4)$$

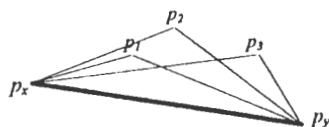


Figura 1. Áreas dos triângulos formados pelos pontos p_x , p_i e p_y , $i=1,2,3$.

Como a Figura 1 ilustra, a base dos triângulos $p_x p_y p_i$ é a mesma, assim a altura do triângulo define a diferença entre as áreas. A rede neural busca a combinação de m vértices com a menor diferença em área.

O segundo método inspirou-se na estratégia proposta por Wu e Leou (1993) no algoritmo de aproximação poligonal denominado *Minimum Area Deviation Polygonal Approximation* (MADPA). Wu e Leou propuseram minimizar a diferença entre as áreas de união e de intersecção, do polígono inicial com o polígono de aproximação. Tal diferença foi calculada somando-se todas as áreas dos polígonos cujos vértices são os pontos dados da curva entre quaisquer p_x e p_y . Estas áreas não pertencem a área de intersecção do polígono inicial com o de aproximação. Resumidamente, pode-se dizer que o MADPA minimiza a diferença de áreas entre o polígono inicial e o polígono de aproximação.

Neste trabalho, emprega-se a diferença de área mencionada para se construir a matriz de conexões H . A rede neural deve achar uma combinação de vértices minimizadoras, em geral mínimos locais, das diferenças de áreas.

As várias áreas de diferenças são calculadas da seguinte maneira:

Seja um polígono de n vértices cujas coordenadas são (x_i, y_i) , $i, j=1,2,\dots,n$. A área deste polígono pode ser calculado utilizando-se a seguinte equação:

$$\text{Área} = \frac{1}{2} \sum_{j=1}^n (x_j y_{j+1} - y_j x_{j+1}) \quad (6)$$

onde: $y_{n+1} = y_1$; $x_{n+1} = x_1$.

Se todos os pontos de vértices estão de um único lado do segmento de reta que une (x_i, y_i) a (x_n, y_n) o cálculo da área é realizado de maneira direta. Porém quando um polígono considerado possui intersecções de arestas, como no caso ilustrado na Figura 2, a equação (6) deve ser

aplicada diferentemente. Nestes casos, aplica-se a equação (6) para cada um dos "subpolígonos", e soma-se as áreas dos "subpolígonos" para achar a área total do polígono.

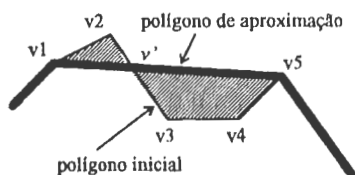


Figura 2. Área do polígono ($v1-v2-v3-v4-v5$) com intersecção de arestas é dada pela soma da área de $v1, v2, v'$ mais a área de $v', v3, v4, v5$.

O algoritmo ganha em desempenho em muitos casos, ao custo do aumento em complexidade. Este último se deve à necessidade de identificar pontos de intersecção de arestas, (vértice v' na Figura 2), que não são pontos dados da curva.

Variando as formas de calcular a matriz de conexões H , a solução do problema também pode mudar, ainda assim a nova solução pode não ser a melhor. Neste caso, é interessante implementar um "professor" que supervisione o treinamento para se chegar a uma solução desejada.

Esta terceira proposta do artigo, foi implementada da seguinte forma: após chegar em uma solução, através do uso de uma das três formas de calcular o H , se o usuário desejar melhorar ou adequar tal solução, ele troca alguns dos pontos selecionados como vértices do polígono de aproximação. A partir daí, a rede é treinada através da Regra Delta, alterando a matriz H , para se obter a saída desejada.

Seja $T_{x,i}$ o estado desejado da unidade (x,i) , e $E_{x,i}$ o erro, ou seja, a diferença entre o estado desejado e o estado real da unidade, então:

$$E_{x,i} = T_{x,i} - V_{x,i} \quad (7)$$

Aplicando-se a Regra Delta, tem-se:

$$\Delta h_{x,y} = \varepsilon E_{x,i} V_{y,i+1} \quad (8-a)$$

$$\Delta h_{y,x} = \varepsilon E_{x,i} V_{y,j-1} \quad (8-b)$$

onde ε é a taxa de aprendizagem.

Baseado nos estados desejados a rede adapta os componentes da matriz H para chegar à saída alvo.

Resume-se o algoritmo de aproximação poligonal empregando Rede Competitiva de Hopfield da seguinte forma:

(i) **Entrada:** Pontos de uma curva $P = \{p_1, p_2, \dots, p_n\}$ no sentido horário e número de vértices do polígono de aproximação m .

(ii) **Saída:** Estado final da rede, onde as unidades ativas representam os vértices do polígono de aproximação $Q^{(m)} = \{q_1, q_2, \dots, q_m\}$

(iii) **Método:**

a) Regra de aprendizagem¹: construção da matriz de conexões H , formada pelos desvios ponto-corda como definido em (4).

b) Estado inicial da rede: ajuste dos estados iniciais das unidades como m pontos, o mais equidistante possível, ao longo da curva.

c) Regra de propagação: cálculo da soma ponderada para cada unidade (x, i) ao longo da coluna i , usando a equação (3).

d) Regra de ativação: aplicação da regra "winner-take-all" da equação (2) para se obter os novos estados para cada unidade da coluna.

e) Repetição dos passos c) e d) até não haver mudanças de estado na rede.

f) Geração dos vértices do polígono de aproximação q_1, q_2, \dots, q_m , para os quais as unidades estão ativas.

Os resultados da Rede Competitiva de Hopfield são satisfatórios em muitos casos. Em particular, se o número de vértices do polígono de aproximação for maior que a metade do número de pontos dados da curva, se a curva não tiver muitos pontos de mudança de direção, se as curvas não diferirem muito de polígonos regulares, ou se as curvas forem figuras convexas, as aproximações são frequentemente boas. Os autores deste trabalho verificaram que quando ocorre um caso distinto dos mencionados acima, os resultados deixam a desejar em várias situações.

3. Melhorando o Aprendizado da Rede

Para melhorar o desempenho do algoritmo, as unidades ativas no final do processo devem ser outras, ao menos em parte. Logo, investigou-se o efeito de mudanças no mecanismo das regras de ativação e de aprendizagem.

A variação da regra de ativação (2) no estado final da rede mostrou-se infrutífera pois a maneira como a matriz de conexões H é construída não deixa margens a ajustes finos de escolha de vencedores de competições. A segunda alternativa, ajuste da matriz de conexões H , mostrou-se ser a estratégia mais adequada. As alterações introduzidas são apresentadas a seguir.

A matriz de conexões foi modificada de três maneiras diferentes. Nas duas primeiras, a maneira de construir a matriz foi mudada e na terceira opção, a matriz foi modificada depois de construída.

A primeira estratégia utiliza a soma das distâncias ponto-corda ao invés da distância máxima. O objetivo é mimizar tal somatório em lugar da maior das distâncias. O segundo método emprega as áreas dos polígonos formados pelas diferenças entre o polígono inicial² e o polígono de aproximação. Tais áreas devem ser minimizadas. O terceiro método utiliza um método de aprendizagem supervisionado para adaptar a matriz de conexões H . Neste caso, o algoritmo original é executado gerando uma primeira solução. Em seguida, o usuário aponta que vértices deseja mudar no polígono de aproximação resultante. De posse deste alvo o programa modifica a matriz H empregando a Regra Delta.

Nas alterações realizadas foi sempre preservada a arquitetura original da Rede Competitiva de Hopfield. Toda mudança se verifica na relação entre as várias unidades da rede numa mesma coluna.

No primeiro caso, os componentes da matriz H deixam de ser o valor máximo das distâncias perpendiculares dos pontos da curva até a corda entre os pontos p_x e p_y . Propõe-se o uso das somas de todas aquelas distâncias. A equação (4) é re-escrita da seguinte forma:

$$h_{xy} = \sum_{p \in P, p \neq p_x, p_y} d(p, \overline{p_x p_y}) \quad (5)$$

O cálculo de h_{xy} utiliza a soma das distâncias como medida aproximada de diferença de área entre o polígono de aproximação e o polígono inicial. Esta diferença é proporcional a soma das áreas dos triângulos de vértices p_x, p_i e $p_y, i=1, 2, 3$.

¹ É conveniente salientar que esta regra tem processo construtivo distinto daqueles em redes neurais.

² Polígono inicial é aquele cujos vértices são todos os pontos dados da curva.

Resumidamente, as mudanças no método da Rede Competitiva de Hopfield são:

(i) Estratégia 1: Os termos h_{xy} da matriz de conexões H são calculados pela equação (5).

(ii) Estratégia 2: Os termos h_{xy} da matriz de conexões H são calculados pela equação (6).

(iii) Estratégia 3: A matriz H é modificada empregando-se as equações (7) e (8) para se chegar a uma resposta desejada.

Cada uma destas estratégias foi testada em casos onde o algoritmo inicialmente descrito não teve desempenho desejado. Resultados ilustrativos são mostrados e comentados a seguir.

4. Simulações

Os métodos propostos juntamente com o método original foram simulados em um microcomputador do tipo IBM-PC, com microprocessador Intel-486, através de programas escritos em linguagem C.

Os métodos são ilustrados em quatro curvas, onde o algoritmo original apresenta desempenho inadequado. As figuras 3 a 6 mostram os resultados obtidos pela aplicação dos diferentes métodos. O item (a) mostra o método original

utilizando máxima distância, o item (b) o método modificado usando soma das distâncias, o item (c) o método modificado empregando soma das áreas e no item (d) o resultado da aproximação obtida pelo método original retreinado por Regra Delta. Mostra-se polígonos de aproximação diferentes para valores de n e m dados. Ainda são apresentados o número de iterações (nit) para o qual a rede converge, o erro quadrático (EQ) e o erro máximo (EM). No caso da Regra Delta menciona-se o número de pontos trocados na tela (np).

Os erros quadrático e máximo são definidos como:

$$\text{Erro Quadrático: } EQ = \sum_{i=1}^n (e_i)^2$$

$$\text{Erro Máximo: } EM = \max_{i=1,2,\dots,n} \{e_i\}$$

onde e_i é a distância perpendicular de um ponto p_i da curva original até a corda considerada no polígono de aproximação. Os erros calculados medem a distância dos pontos na tela.

Observe que de maneira geral, nestes casos, os métodos propostos neste trabalho melhoram a aproximação poligonal.



(a)



(b)



(c)



(d)

Figura 3. Aproximação poligonal para $n = 20$ e $m = 10$. (a) nit=2, EQ=2705.9, EM=31.2; (b) nit=1, EQ=404.5, EM=13.9; (c) nit=1, EQ=404.5, EM=13.9; (d) np=1, EQ=1119.2, EM=24.2.



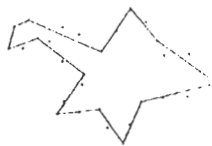
(a)



(b)



(c)



(d)

Figura 4. Aproximação poligonal para $n = 30$ e $m = 13$. (a) nit=2, EQ=3015.3, EM=36.3; (b) nit=2, EQ=1105.2, EM=13.8; (c) nit=1, EQ=3336.3, EM=39.3; (d) np=2, EQ=1230.6, EM=13.8.

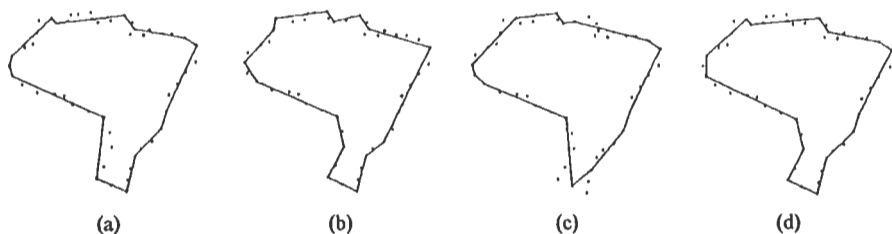


Figura 5. Aproximação poligonal para $n = 50$ e $m = 15$. (a) nit=2, EQ=1530.8, EM=17.3; (b) nit=4, EQ=804.6, EM=9.3; (c) nit=4, EQ=2344.4, EM=22.4; (d) np=3, EQ=1069.3, EM=10.6.

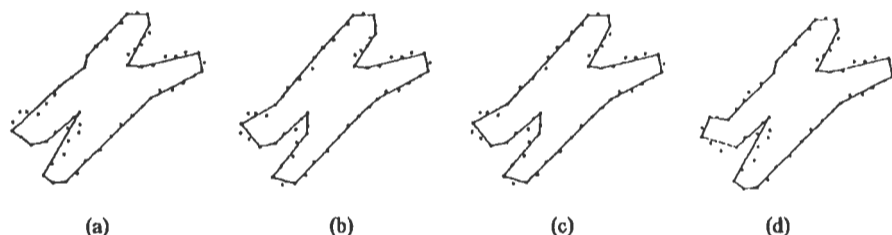


Figura 6. Aproximação poligonal para $n = 60$ e $m = 18$. (a) nit=2, EQ=869.1, EM=11.5; (b) nit=2, EQ=542.5, EM=7.8; (c) nit=2, EQ=520.3, EM=7.8; (d) np=4, EQ=760.6, EM=11.5.

5. Discussão

O problema de aproximação poligonal tem

$$C_m^n = \frac{n!}{m!(n-m)!}$$

possíveis combinações dos m

vértices do polígono de aproximação, dentre os n pontos dados da curva original. Este artigo propõe variações no modelo de Rede Competitiva de Hopfield para aproximação poligonal, visando diminuir suas limitações.

As três modificações propostas são referentes a alterações na matriz de conexões H . Esta é construída de duas maneiras diferentes e modificada através de aprendizagem supervisionada.

As simulações sugerem que os métodos propostos neste artigo melhoram o desempenho do método original quando m é menor do que a metade de n , quando a curva tem muitos pontos de mudança de direção, quando a curva difere muito de polígonos regulares e/ou quando a curva não é figura convexa. Quando m se aproxima de n , as soluções encontradas pelos diferentes métodos coincidem.

Entende-se que o melhor desempenho dos dois primeiros métodos propostos, nos casos críticos do método original, se deve ao fato de que as

medidas de desvio utilizadas nas novas estratégias avaliam melhor a proximidade entre o polígono de aproximação e a curva original. Na terceira estratégia, a aprendizagem "batch" inicial é aperfeiçoado por aprendizagem "on-line".

Referências

- Chung, P. C., Tsai, C. T., Chen, E. L., Sun, Y. N. (1994). Polygonal Approximation Using a Competitive Hopfield Neural Network. *Pattern Recognition* 27, 1505-1512.
- Hopfield, J. J. (1982). Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proc. Natl. Acad. Sci. U.S.A.* 79, 2554-2558.
- Rumelhart, D.E. e Zisper, D. (1985). Feature Discovery by Competitive Learning. *Cognitive Science*, 9, 75-112.
- Widrow, B. e Hoff, M.E. (1960) Adaptive Switching Circuits. 1960 IRE WESCON Convention Record, New York : IRE, 96-104.
- Wu, J. S. e Leou, J. J. (1993). New Polygonal Approximation Schemes for Object Shape Representation. *Pattern Recognition* 26, 471-484.