# USO DE REDES ARTIFICIAIS DE NEURÔNIOS LINEARES PARA O TREINAMENTO E O RECONHECIMENTO DE CARACTERES COM ROTAÇÃO

**Gilberto Arantes Carrijo**
Deptº de Engª Elétrica - UFU
38.400 - Uberlândia - MG

**RESUMO** - O treinamento de uma rede artificial de neurônios lineares de duas camadas (ADALINE) é discutido neste artigo. Considerações sobre o problema da convergência é feito baseando-se na regra delta de Widrow-Hoff e uma solução exata para o problema é apresentada.

## 1. INTRODUÇÃO

As redes artificiais de neurônios (RAN) estão sendo hoje usadas em várias áreas tais como, reconhecimento de sinais de voz, robótica, reconhecimento de caracteres etc.. A eficiência de tais redes é relativamente alta em todas estas áreas, mas vários problemas ainda existem, sendo um deles o da lentidão da convergência. O algoritmo da propagação retrógrada (back-propagation) de Rumelhart e autores [1] está sendo largamente usado no treinamento das RANs. Este algoritmo possui limitações muito grandes, a principal delas é o uso do fator de aprendizado baseando se na experiência que os pesquisadores adquirem ao longo de seus trabalhos, não existindo uma maneira precisa de determiná-lo.

As principais pesquisas nesta área estão direcionadas na tentativa de aumentar a velocidade de convergência das RANs. A solução deste problema é de vital importância, pois pode reduzir enormemente as horas de processamento. Temos trabalhado neste sentido na área de reconhecimento de voz e apenas como título de informação podemos em alguns casos reduzir por um fator de 10 o número de interações necessárias à convergência.

Outro assunto largamente discutido é o problema da solução se dirigir para um mínimo local ou global. Este tipo de problema ainda não foi solucionado, mas após trabalhar por algum tempo na área, podemos arriscar em dizer que esta não é uma situação crítica e crucial, pois o que interessa é saber se a RAN converge ou não para uma situação que fica dentro de uma margem de erro considerada satisfatória.

Como o trabalho aqui proposto é o de reconhecimento de caracteres, passaremos agora a analisar mais de perto o problema baseado na técnica das redes artificiais de neurônios. A maioria dos processos de reconhecimento de caracteres baseados nos métodos tradicionais estão funcionando de maneira eficiente. Surge então a pergunta. Porque usar as RANs? Ora estas técnicas tradicionais não executam o processamento paralelo, e as vezes são muito lentas, enquanto as RANs usam o processamento paralelo, aumentando assim enormemente a velocidade de reconhecimento. Outros fatores encorajam o uso das RANs mas apenas este já é suficiente para justificar a pesquisa na área.

Um dos primeiros trabalhos do uso das redes artificiais de neurônios no reconhecimento de caracteres foi desenvolvido por Widrow [2]. O referido autor faz uso da ADALINE (adaptive linear neuron). Widrow [2] construiu uma máquina com vários potenciômetros constituindo assim a RAN que era ajustada variando estes potenciômetros até a convergência ser atingida. A meu ver foi um trabalho cansativo, mas os resultados foram excelentes. A partir daí as pesquisas na área ficaram paralizadas, quando então com resurgimento da pesquisa na área das RANs Widrow [3] publicou outro artigo, mas apenas dando idéias do que poderia ser feito nesta área. Os assuntos abordados por Widrow [3] estão relacionados aos problemas de rotação, translação, escalamento etc de caracteres. Algumas novas estruturas foram sugeridas no seu trabalho, mas as vezes muito complexas para implementações. Isto talvez justifique o não surgimento de novos trabalhos seguindo suas idéias. Neste trabalho procuraremos analisar o problema do reconhecimento de caracteres quando se faz rotações nos mesmos. O trabalho não esgota todos os desenvolvimentos na área, mas apenas dará uma contribuição ao assunto.

## 2. NEURÔNIOS LINEARES ADAPTATIVOS (ADALINE)

As noções de redes artificiais de neurônios adaptativos iniciaram com Widrow [2] em 1962. Trabalhando com hardware Widrow [2] desenvolveu um equipamento que fazia aprendizado de caracteres usando uma matriz 4x4. A mesma rede foi usada por Widrow [3] em seu trabalho mais recente. A rede linear adaptativa (ADALINE) estudada por Widrow [2] é constituída por apenas duas camadas, a camada de entrada e a camada de saída como mostra a figura 1.

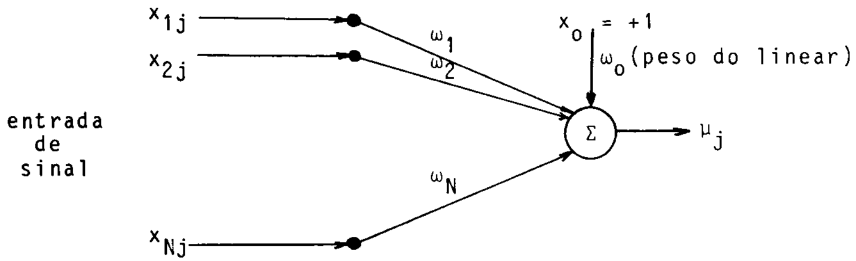



Figura 1. Rede artificial de neurônios lineares adaptativos (ADALINE).

Esta é a rede mais simples que se pode ter. Os sinais de entrada são designados pelos vetores $\vec{x}_j$, onde j representa o padrão.

$$[\vec{x}j]^T = (x_o \; x_{1j} \; x_{2j} \; \cdots \; x_{nj}) \tag{1}$$

Os pesos são designados por $\vec{\omega}$,

$$[\vec{\omega}]^T = (\omega_o \; \omega_1 \; \omega_2 \; \cdots \; \omega_n) \tag{2}$$

A entrada $x_o$ é fixada em 1 e representa o limiar. A saída da rede de Widrow [2] é representada por um neurônio de decisão tipo binária ($\pm$ 1). Com esta rede consegue se treinar duas classes de padrões ou seja dois caracteres pois a decisão é binária e apresenta apenas duas opções.

Todo o processo começa com o treinamento da rede, ou seja as entradas são apresentadas e os pesos são calculados usando-se algoritmos de treinamento. Consideremos como exemplo dois padrões C e T representativos de duas classes como mostrado na figura 2.

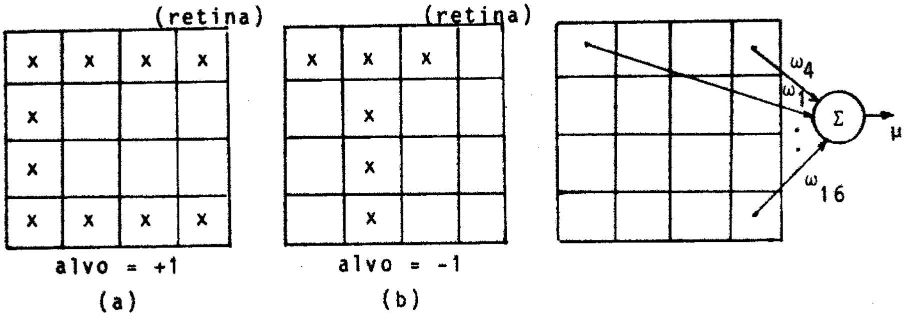



Figura 2. Dois padrões C e T apresentados nas retinas.

A cada elemento da retina está associado um peso $\omega_n$ da rede. Quando se apresenta um elemento pertencente uma classe associamos a sua saída ao valor $\pm$ 1 (1 para C e -1 para T) que se chama de alvo. De maneira geral a saída da rede da figura 1 é dada por:

$$\mu_j = \sum_{i=1}^{N} x_{ij} \quad \omega_i \tag{3}$$

N é o número de entradas

j representa o padrão, para o caso da ADALINE j = 1, 2.

O erro é dado pela expressão:

$$\varepsilon = \sum_{j=1}^{M} \frac{1}{2}(\mu_j - t_j) \tag{4}$$

Onde $t_j$ são os alvos. M é o número de padrões total usados no treinamento.

A determinação dos pesos é feita por algoritmos de treinamento que serão vistos a seguir.

## 3. USO DA REGRA DELTA DE WIDROW-HOFF PARA UM NEURÔNIO NA ROTAÇÃO DE CARACTERES

As redes artificiais de neurônios são utilizadas inicialmente para treinamento ou seja determinação dos pesos, e logo após para classificar os elementos de acordo com a classe. Vá rias técnicas são usadas no treinamento, sendo a princial delas a propagação retrógrada (back propagation). A regra delta de Widrow-Hoff [3] segue o mesmo princípio da propagação retrógrada apresentada por Rumelhart [1] e pode ser dada por:

$$\vec{\omega}_{k+1} = \vec{\omega}_k + \frac{\alpha}{|\vec{x}_k|} \cdot \varepsilon \, \vec{x}_k \qquad (5)$$

onde $\omega_k$ é o valor do peso na interação k, $\omega_{k+1}$ é o valor do peso na interação k+1, $\alpha$ é o fator de aprendizado e $\varepsilon$ é o erro, isto é, o valor entre a saída e o alvo. O fator de aprendizado está no intervalo:

$$1,0 > \alpha > 0,1 \qquad (6)$$

A regra delta nada mais é do que a regra de otimização de funções empregando-se o método do gradiente [4].

Nosso principal objetivo será aplicar a regra de Widrow-Hoff no treinamento da rede.

Consideremos inicialmente duas classes, sendo que cada classe possue quatro padrões cada com uma rotação de $90^o$ comparado ao outro, como mostra a figura 3.

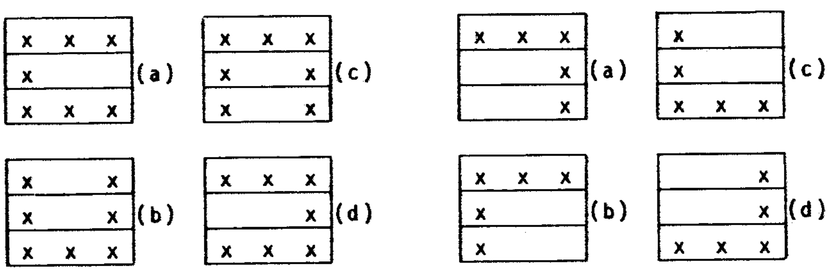



Figura 3. Padrões usados no treinamento da rede.

Os oito padrões apresentados à rede da figura 1 com oito entradas e nove pesos dada por $\omega_1$, $\omega_2$ ... $\omega_8$, $\omega_9$ após o treinamento usando se 100 interações, $\alpha = 0,5$ obtivemos um $\varepsilon < 2,5 \times 10^{-5}$, e os pesos calculados são dados por:

| | | |
|---|---|---|
| $\omega_1$ 0,1824 | $\omega_2$ 0,8176 | $\omega_3$ 0,1824 |
| $\omega_4$ 0,8176 | $\omega_5$ 1,3648 | $\omega_6$ 0,8176 |
| $\omega_7$ 0,1824 | $\omega_8$ 0,8176 | $\omega_9$ 0,1824 |

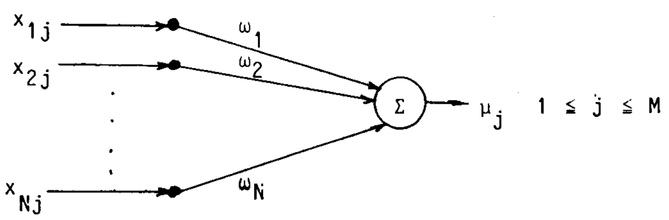
Nota-se que $\omega_1 = \omega_3 = \omega_7 = \omega_9 = 0,1824$ e $\omega_2 = \omega_4 = \omega_8 = \omega_6 = 0,8176$. Este problema é totalmente simétrica pois quando se roda de $90^o$ a figura 3.a, obtém se a figura 3.b e o peso representado por $\omega_3$ passa para $\omega_1$, $\omega_1$ para $\omega_7$ e $\omega_7$ para $\omega_9$. O mesmo acontece com $\omega_2$, $\omega_4$, $\omega_8$ e $\omega_6$. Como o alvo é o mesmo para cada classe, estes pesos podem ser iguais, mas não obrigatoriamente iguais como será visto posteriormente.

## 4. SOLUÇÃO EXATA PARA A REDE ARTIFICIAL LINEAR DE NEURÔNIOS

O principal problema hoje enfrentado no processo de treinamento de uma RAN está relacionado ao processo de convergência da rede. O processo de convergência ainda continua sendo estudado à luz da teoria da propagação retrógrada, e alguns pesquisadores tentam obter resultados sem fazerem uma análise matemática sobre o assunto. É comum ouvirmos frazes, tais como, "a rede não converge".

Tentaremos neste parágrafo analisar o problema para o caso de uma RAN, como aquela apresentada na figura 1.

Considerando então uma ADALINE de N entradas, uma saída assumindo valores $\pm$ 1 e M padrões divididos em duas classes, podemos escrever o sistema de equações (7).




$$\sum_{i=1}^{N} x_{ij} \cdot \omega_i = t_j \qquad j \leq M \qquad (7)$$

Temos assim um sistema de M equações e N incógnitas. Se $M > N$, não haverá solução, por outro lado se $M < N$, de maneira geral teremos um número infinito de soluções.

Como simples exemplo consideremos os padrões da figura 3.a.

| | | |
|---|---|---|
| X | X | X |
| X | | |
| X | X | X |

alvo = 1

| | | |
|---|---|---|
| X | X | X |
| | | X |
| | | X |

alvo = -1

Para este caso obtemos as equações,

$$\omega_1 + \omega_2 + \omega_3 + \omega_4 - \omega_5 - \omega_6 + \omega_7 + \omega_8 + \omega_9 = 1$$

$$\omega_1 + \omega_2 + \omega_3 - \omega_4 - \omega_5 + \omega_6 - \omega_7 - \omega_8 + \omega_9 = -1$$

Se atribuírmos valores a $\omega_1 = \omega_2 = \omega_3 = \omega_4 = \omega_5 = \omega_6 = \omega_7 = 0$,

Chegaremos ao sistema,

$$+ \omega_8 + \omega_9 = 1 \qquad - \omega_8 + \omega_9 = -1 \tag{8}$$

cuja solução será;

$$\omega_8 = 1 \qquad \omega_9 = 0 \tag{9}$$

## 5. SOLUÇÃO EXATA DA REDE ARTIFICIAL DE NEURÔNIOS LINEARES USANDO A MINIMIZAÇÃO DA FUNÇÃO ERRO

A determinação dos pesos é feita minimizando a função erro. Todos os algoritmos trabalham neste sentido, fazendo com que a função erro atinge o valor mínimo. Considerando uma rede RAN como aquela da figura 1 com N entradas e um total de M padrões de entrada o erro será dado por:

$$\varepsilon = \frac{1}{2}(\mu_1 - t_1)^2 + \frac{1}{2}(\mu_2 - t_2)^2 + \ldots \frac{1}{2}(\mu_M - t_M)^2 \tag{10}$$

Onde as saídas são:

$$\mu_1 = \sum_{i=1}^{N} x_{i1}\,\omega_i \;;\; \mu_2 = \sum_{i=1}^{N} x_{i2}\,\omega_i \;;\; \mu_M = \sum_{i=1}^{N} x_{iM}\omega_i \tag{11}$$

e $t_1, t_2, \ldots, t_M$ são os alvos que no caso da ADALINE de Widrow [2] assume apenas os valores $\pm 1$, mas no caso pode assumir qualquer valor.

A determinação dos pesos é feita minimizando a função erro com respeito aos pesos.

Derivando (10) e (11) com respeito a um peso genérico j, obteremos:

$$\frac{\partial \varepsilon}{\partial \omega_j} = \frac{\partial \varepsilon}{\partial \mu_j} \cdot \frac{\partial \mu_j}{\partial \omega_j} = 0 \quad , \qquad 1 \le j \le N \tag{12}$$

$$\frac{\partial \varepsilon}{\partial \omega_j} = (\mu_1 - t_1)x_{j1} + (\mu_2 - t_2)x_{j2} + \ldots (\mu_M - t_M)x_{jM} = 0$$

Obteremos o sistema de equações,

$$\sum_{i=1}^{N}(x_{i1}\ x_{11} + x_{i2}\ x_{12} + \ldots, x_{iM}\ x_{1M})\omega_i = \sum_{i=1}^{N} t_i x_{1i}$$

$$\vdots \tag{13}$$

$$\sum_{i=1}^{N}(x_{i1}\ x_{N1} + x_{i2}\ x_{N2} + \ldots, x_{iM}\ x_{NM})\omega_i = \sum_{i=1}^{N} t_i x_{Ni}$$

A solução do sistema (13) fornecerá os N pesos da RAN. Para sermos claros, ilustrarmos duas situações onde $N = M$, e $N < M$. Considerando inicialmente $N = M = 2$, e os vetores $\vec{x}_1$ e $\vec{x}_2$ dados por,

$$\vec{x}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \qquad \vec{x}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{14}$$

alvos dados por,

$$t_1 = 0 \qquad t_2 = 1$$

As saídas serão,

$$\mu_1 = \omega_1 + 2\omega_2 \qquad \mu_2 = \omega_1 + \omega_2 \tag{15}$$

Derivando a função erro teremos,

$$\frac{\partial \varepsilon}{\partial \omega_1} = (\omega_1+2\omega_2)+(\omega_1+\omega_2-1) = 0 \qquad \frac{\partial \varepsilon}{\partial \omega_2} = (\omega_1+2\omega_2).2+(\omega_1+\omega_2-1) = 0 \tag{16}$$

A solução do sistema de equações (16) é,

$$\omega_1 = 2 \qquad \omega_2 = -1 \tag{17}$$

As saídas $\mu_1$ e $\mu_2$ serão de acordo com (15),

$$\mu_1 = 0 \qquad \mu_2 = 1 \tag{18}$$

Temos então que a solução será exata e o erro será nulo. Considerando agora o caso dos padrões $\vec{x}_1$, $\vec{x}_2$, $\vec{x}_3$,

$$\vec{x}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad , \quad \vec{x}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad , \quad \vec{x}_3 = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \tag{19}$$

$$t_1 = 0 \quad , \quad t_2 = 1 \quad , \quad t_3 = 2$$

As saídas são,

$$\mu_1 = \omega_1 + 2\omega_2 \; ; \; \mu_2 = \omega_1 + \omega_2 \; ; \; \mu_3 = 2\omega_1 + \omega_2 \tag{20}$$

$$\frac{\partial\varepsilon}{\partial\omega_1} = (\omega_1+2\omega_2)+(\omega_1+\omega_2-1)+(2\omega_1+\omega_2-2).2=0 \; ; \; \frac{\partial\varepsilon}{\partial\varepsilon_2} = (\omega_1+2\omega_2).2+(\omega_1+\omega_2-1) +$$

$$+(2\omega_1+\omega_2-2) = 0 \tag{21}$$

A solução é,

$$\omega_1 = -15 \qquad \omega_2 = 13 \tag{22}$$

As saídas $\mu_1$ e $\mu_2$ são,

$$\mu_1 = 11 \qquad \mu_2 = -3 \qquad \mu_3 = -19 \tag{23}$$

O erro será

$$\varepsilon = \frac{1}{2}(11)^2 + \frac{1}{2}(-3)^2 + \frac{1}{2}(19)^2 = 248,5 \neq 0$$

Nota-se neste caso que o erro não é nulo. Generalizando, podemos dizer que quando $N < M$, os pesos calculados pelo processo de minimização da função erro nos produzirá uma solução que será o mínimo da função erro, mas este mínimo não será nulo. Este resultado é muito interessante, pois elimina aquela dúvida se a rede converge ou não para o alvo. Resumidamente dizemos que,

se $N < M$ não há convergência para o alvo

se $N \geq M$ há convergência para o alvo

onde $N$ é o número de entradas ou peso e $M$ é o número total de padrões.

Experimentos usando a solução exata e o algoritmo de Widrow-Hoff foram feitos para várias redes. Dois exemplos são mostrados a seguir para os casos $N = 8$, $M = 16$ e $N = 8$, $M = 12$.

| | | | |
|---|---|---|---|
| 0,2096 | 0,2672 | 0,0984 | 0,2317 |
| 0,0893 | -0,0751 | -0,0363 | 0,2826 |
| 0,3135 | -0,0210 | -0,0841 | 0,0982 |
| 0,1945 | 0,1114 | 0,3065 | 0,1944 |

$N = 8$
$M = 16$

Quando $N \neq M$ não ha simetria perfeita dos pesos.

| | | |
|---|---|---|
| -0,5009 | 0,2344 | 0,2683 |
| 0,5022 | 0,2406 | 0,2513 |
| 0,2592 | 0,2491 | -0,4984 |
| 0,2828 | 0,2136 | -0,4964 |

N = 8
M = 12

## 6. CONCLUSÕES

O estudo de uma rede artificial de neurônios lineares foi feito usando a minimização de funções, e uma solução exata foi encontrada para o problema, não sendo necessário assim usar os algoritmos de "back-propagation" ou a regra de Widrow-Hoff. Quando o número de padrões usados no treinamento for menor ou igual ao número de pesos ($M \leq N$) a solução converge para o ponto de energia nulo. Caso contrário a energia não será nula no ponto de mínimo. A rotação produzirá pesos simétricos de acordo com a simetria da rotação, mas esta simetria será perfeita apenas quando o número de padrões for igual ao número de pesos ($M = N$).

## BIBLIOGRAFIA

[1] D.E. Rumelhart e J. L. McClelland, Parallel Distributed Processing, Vol. I e II. Cambridge, M.A.: M.I.T. Press, 1986.

[2] B. Widrow, "Generalization and information storage in networks of adaline neurons", em Self-organization Systems 1962, M.C. Yovitz, G.T. Jacobi and G.D. Goldstein, Eds. Washington, D.C.: Spartan Books, 1962, pp. 435-461.

[3] B. Widrow, R.G. Winter e R.A. Baxter, "Layered Neural Nets for pattern recognition", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 36, nº 7, July 1988, pp. 1109-1118.

[4] D. M. Himmelbau, Applied Nonlinear Programming, McGraw-Hill, New York (1972).