

## 1º Congresso Brasileiro de Redes Neurais

Escola Federal de Engenharia de Itajubá  
Itajubá. 24 a 27 de outubro de 1994

### Reconhecimento de Caracteres Manuscritos Utilizando Redes Neurais

*\*Edna L. Flóres, \*Eder N. Rezende, \*Gilberto A. Carrizo e \*\*João B. T. Yabuti.*

\*Departamento de Engenharia Elétrica  
Universidade Federal de Uberlândia  
38.400-902 - Uberlândia - MG.

\*\*Departamento de Comunicações  
Universidade Estadual de Campinas  
13.081 - 970 - Campinas - SP.

**Resumo** - Este artigo trata do reconhecimento de caracteres escrito à mão através de redes neurais. Para chegarmos a um resultado satisfatório os caracteres são normalizados, tornando-se insensíveis à escala, translação e rotação. O artigo consta de 4 tópicos principais, onde no primeiro trata-se das técnicas de reconhecimento; no segundo é exposta a técnica utilizada para tornar os caracteres invariantes à escala, translação e rotação; no terceiro tópico trata-se brevemente dos algoritmos utilizados para o reconhecimento e descreve-se os resultados obtidos. No quarto tópico são feitas conclusões a respeito dos resultados.

**Palavras Chaves** - Caracteres Manuscritos, Escala, Translação, Rotação, Reconhecimento.

#### 1 - INTRODUÇÃO

Vários centros de pesquisas e Universidades estão tentando desenvolver técnicas de reconhecimento de caracteres manuscritos que tenham um nível de desempenho praticamente igual ao do homem. O motivo disto é a grande importância nas aplicações científicas e econômica, tais como: reconhecimento de código de endereçamento postal, leitura automática de cheques bancários, sistemas de tarifação telefônica, sistemas de leitura automática para cegos, etc. A obtenção de um software que consiga fazer o papel do

homem pode representar uma grande economia de tempo e um bom ganho para as empresas. O reconhecimento de caracteres possui 3 passos principais: aquisição da imagem, segmentação, e finalmente classificação dos caracteres. No processo de aquisição da imagem os caracteres são obtidos através de um processo óptico qualquer, tal como em "scanner". A Segmentação dos caracteres é o processo no qual os caracteres que aparecem conectados são separados. Classificação é o processo pelo qual os caracteres são comparados a padrões já existentes para o seu reconhecimento.

## 2-ESCALA, TRANSLAÇÃO E ROTAÇÃO [1]

O reconhecimento de caracteres manuscritos utilizando redes neurais não apresenta resultados satisfatórios, se usarmos caracteres no treinamento diferentes daqueles usados no reconhecimento. Ocorre o mesmo quanto a translação, se a rede for treinada com caracteres em uma posição e no reconhecimento estes estiverem situados em outras posições. Quanto a rotação, isso também ocorre, se os caracteres são treinados inclinados com um determinado ângulo e no reconhecimento estes estiverem com outros ângulos de inclinação. Em todos os casos o reconhecimento fica bastante prejudicado. O algoritmo [1] desenvolvido para tornar o caracter invariante à escala, translação e rotação segue os seguintes passos:

### INVARIÂNCIA A ESCALA

O algoritmo permite que o caracter seja ampliado ou reduzido.

Define-se:

$n_l$  - número de linhas da imagem.

$n_c$  - número de colunas da imagem.

$n_{ln}$  - número de linhas da imagem na nova escala.

$n_{cn}$  - número de colunas da imagem na nova escala.

$e_l$  - fator de escala das linhas.

$e_c$  - fator de escala das colunas.

$e_f$  - fator de escala final da imagem.

$$e_l = \frac{n_{ln}}{n_l}, \quad e_c = \frac{n_{cn}}{n_c} \quad (1)$$

O fator de escala final que a imagem fica submetida é calculado do seguinte modo:

. Se a diferença entre  $e_l$  e  $e_c$  for menor que 0.4, o  $e_f$  será igual a média entre  $e_l$  e  $e_c$ .

. Se  $e_l$  for maior ou igual a 1 e  $e_c$  menor ou igual a 1, o  $e_f$  será o maior dos dois termos.

. Se  $e_l$  e  $e_c$  forem maior ou igual a 1, o  $e_f$  será o menor dos dois termos.

. Se  $e_l$  for maior ou igual a 2 o  $e_f$  será igual a  $e_c$ , caso contrário será  $e_l$ .

A imagem invariante a escala será:

$$f_E(x_i, y_j) = f(e_f \cdot x_i, e_f \cdot y_j) \quad (2)$$

onde  $f(x_i, y_j)$  fornece o valor do "pixel" da coordenada  $(x, y)$ .

### INVARIÂNCIA A TRANSLAÇÃO

A translação é realizada utilizando a matriz resultante do processo de escala.

Para que o caracter fique insensível a translação é calculado seu centro de gravidade e deslocado este para a origem do novo sistema de coordenadas que é o centro da matriz que contém o caracter.

O centro de gravidade é calculado pela média das coordenadas  $x$  e  $y$  dos "pixels-on", de acordo com as fórmulas abaixo:

$$P = \sum_{i=1}^N \sum_{j=1}^N f_E(x_i, y_j) \quad (3)$$

onde  $P$  é o número de "pixels-on" da matriz do caracter.

O centro de gravidade será:

$$x_{cg} = \frac{1}{P} \sum_{i=1}^N \sum_{j=1}^N f_E(x_i, y_j) \cdot x_i \quad (4.a)$$

$$y_{cg} = \frac{1}{P} \sum_{i=1}^N \sum_{j=1}^N f_E(x_i, y_j) \cdot y_j \quad (4.b)$$

O deslocamento do centro de gravidade do caracter para a origem do novo sistema de coordenadas (centro de gravidade da matriz) é calculado da seguinte maneira:

$$x_{cgm} = x_{cg} - x_o, \quad y_{cgm} = y_{cg} - y_o \quad (5)$$

onde:

$x_o$  - abscissa do centro da matriz do caracter.  
 $y_o$  - ordenada do centro da matriz do caracter.

$x_{cgm}$  - abscissa do centro de gravidade da matriz.

$y_{cgm}$  - ordenada do centro de gravidade da matriz.

A imagem invariante a translação será:

$$f_{ET}(x_i, y_j) = f_E(x_i + x_{cgm}, y_j + y_{cgm}) \quad (6)$$

### INVARIÂNCIA A ROTAÇÃO

A rotação é realizada utilizando a matriz resultante do processo de translação.

O caracter fica invariante à rotação rodando-o de maneira que a direção da variância máxima coincide com o eixo x. A derivação da função é baseada na transformação de Karhunen-Loève que tem sido usada em algumas aplicações [2, 3, 4]. Esta transformação explica o seguinte: dado um conjunto de vetores, o autovetor que corresponde ao maior autovalor da matriz de covariância calculada do conjunto de vetores, pontos na direção da variância máxima [3,4]. Esta propriedade pode ser usada para manter invariância a rotação desde que a detecção da direção da variância máxima revelará o ângulo de rotação. A solução geral para qualquer tamanho de vetores será impraticável. Para vetores de duas dimensões formados pelas coordenadas x e y de "pixels-on", os autovalores são fáceis para calcular e uma fórmula para o autovetor correspondendo ao maior autovalor pode ser derivada da

matriz de covariância 2 x 2. Os parâmetros da rotação são derivados como segue:

Define-se:

$$\begin{bmatrix} m_x \\ m_y \end{bmatrix} = \frac{1}{\sum_{i=1}^N \sum_{j=1}^N f_{ET}(x_i, y_j)} \begin{bmatrix} x_i \\ y_j \end{bmatrix} \quad (7)$$

$$P = \sum_{i=1}^N \sum_{j=1}^N f_{ET}(x_i, y_j) \quad (8)$$

$$T_{xx} = \sum_{i=1}^N \sum_{j=1}^N f_{ET}(x_i, y_j) \cdot x_i^2 \quad (9)$$

$$T_{yy} = \sum_{i=1}^N \sum_{j=1}^N f_{ET}(x_i, y_j) \cdot y_j^2 \quad (10)$$

$$T_{xy} = \sum_{i=1}^N \sum_{j=1}^N f_{ET}(x_i, y_j) \cdot x_i \cdot y_j \quad (11)$$

A matriz de covariância é definida como:

$$C = \frac{1}{P} \left( \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} m_x \\ m_y \end{bmatrix} \right) \left( \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} m_x \\ m_y \end{bmatrix} \right)^T \quad (12)$$

Pode ser simplificada para:

$$C = \left( \frac{1}{\sum_{i=1}^N \sum_{j=1}^N f_{ET}(x_i, y_j)} \sum_{i=1}^N \sum_{j=1}^N f_{ET}(x_i, y_j) \begin{bmatrix} x_i \\ y_j \end{bmatrix} \begin{bmatrix} x_i \\ y_j \end{bmatrix}^T \right) - \left( \begin{bmatrix} m_x \\ m_y \end{bmatrix} \begin{bmatrix} m_x \\ m_y \end{bmatrix}^T \right) \quad (13)$$

Desde que a invariância a translação têm sido mantida, as médias  $m_x$  e  $m_y$  são zero. Além disso, o termo médio na frente da matriz pode ser eliminado desde que ele não muda a direção dos autovetores. Portanto, a matriz de covariância torna-se:

$$C = \begin{bmatrix} T_{xx} & T_{xy} \\ T_{xy} & T_{yy} \end{bmatrix} \quad (14)$$

Finalmente, o seno e cosseno do ângulo de rotação são:

$$\sin \theta = \frac{(T_{yy} - T_{xx}) + \sqrt{(T_{yy} - T_{xx})^2 + 4 \cdot T_{xy}^2}}{\sqrt{(8 \cdot T_{xy}^2 + D) \sqrt{((T_{yy} - T_{xx})^2 + 4 \cdot T_{xy}^2)}}} \quad (15)$$

$$\cos \theta = \frac{2 \cdot T_{xy}}{\sqrt{(8 \cdot T_{xy}^2 + D) \sqrt{((T_{yy} - T_{xx})^2 + 4 \cdot T_{xy}^2)}}} \quad (16)$$

onde :

$$D = 2 \cdot (T_{yy} - T_{xx})^2 + 2 \cdot (T_{yy} - T_{xx}) \quad (17)$$

A imagem invariante à rotação será:

$$f_{ETK}(x_i, y_j) = \frac{f_{ET}(\cos \theta \cdot x_i - \sin \theta \cdot y_j, \sin \theta \cdot x_i + \cos \theta \cdot y_j)}{\quad} \quad (18)$$

### 3 - Redes Neurais

#### 3.1 - Algoritmo Back-Propagation

O algoritmo utilizado para o reconhecimento de caracteres manuscritos foi o back-propagation. Este algoritmo [5] é um processo de treinamento iterativo no qual o sinal erro de saída é propagado

posterior através da rede e é usado para modificar os valores dos pesos. A Figura 1 mostra a rede neural de 3 camadas, utilizada para o reconhecimento de caracteres manuscritos.

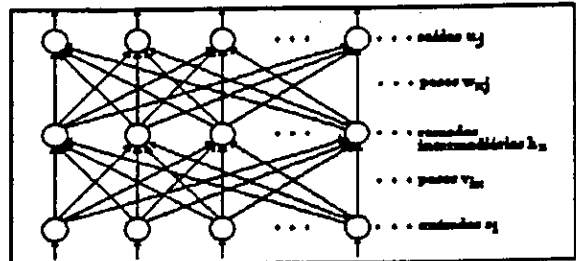


Fig. 1 - Rede neural de 3 camadas.

#### 3.2 - Resultados

Os "pixels" de cada caracter treinado e reconhecido pela rede, invariantes à escala, translação e rotação, são colocados em uma matriz 30x30, logo o número de entradas da rede para cada caracter é 900. O treinamento é realizado para os números de 0 a 9, constituindo assim um arquivo de treinamento, portanto o número de saídas da rede é 10. O número de camadas intermediárias utilizadas para treinamento é 20. A figura 2 mostra os 10 caracteres numéricos de uma pessoa utilizados antes de ficarem invariantes à escala, translação e rotação; a figura 3 mostra os 10 caracteres numéricos de uma pessoa utilizados para treinamento e reconhecimento que são invariantes à escala, translação e rotação.

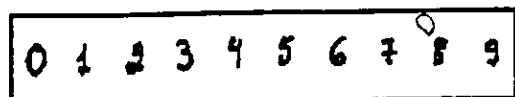


Fig. 2 - Caracteres numéricos de uma pessoa antes de ficarem invariantes à escala, translação e rotação.



Fig. 3 - Caracteres numéricos de uma pessoa depois de ficarem invariantes à escala, translação e rotação.

Foram feitos 3 tipos de testes com caracteres de dimensões 30x30. O primeiro foi feito com os caracteres apenas de uma pessoa e utilizando para treinamento 10 arquivos cada um contendo 10 números decimais para treinamento. Trinta outros arquivos da mesma pessoa e diferentes daqueles usados para o treinamento foram utilizados no reconhecimento. A tabela 1 e a figura 4 apresenta os resultados obtidos.

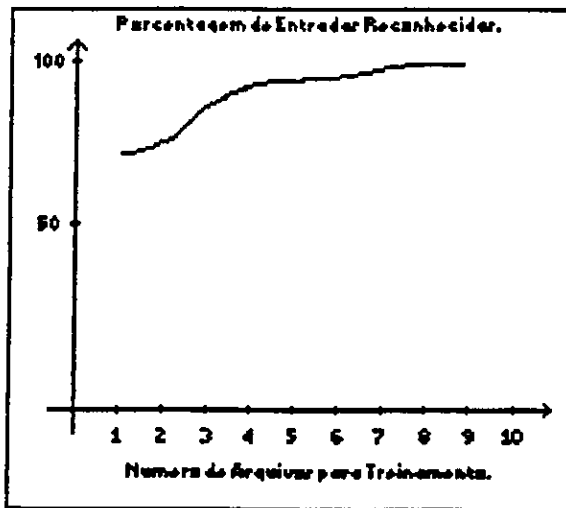


Fig. 4 - Gráfico da porcentagem de entradas reconhecidas x número de arquivos de uma pessoa para treinamento.

Verifica-se que quanto maior o número de arquivos usados no treinamento a porcentagem aumenta. ◯

O segundo teste foi feito com 4 pessoas diferentes. Inicialmente foi utilizado um arquivo contendo 10 números decimais de cada pessoa para o treinamento. No reconhecimento foi utilizado outros 10 arquivos das mesmas pessoas, totalizando 40 arquivos ou um total de 400 caracteres. Neste caso obtivemos um reconhecimento de 56.25%. O mesmo teste foi feito

tomando-se 2, 3 e 4 arquivos de cada uma das 4 pessoas acima mencionadas. A porcentagem de acerto aumentou gradativamente atingindo 71.00% de acerto para o caso de 4 arquivos de cada pessoa. Em todos os testes foram utilizados no reconhecimento 10 arquivos de cada pessoa. Os resultados são mostrados na tabela 2 e figura 5.

Tabela 1 - Treinamento e reconhecimento com arquivos de uma pessoa.

Número de Arquivos para Treinamento	Número de Entradas Reconhecidas	Porcentagem de Entradas Reconhecidas
1	208	69.30%
2	219	73.00%
3	265	83.30%
4	266	88.87%
5	271	90.33%
6	272	90.67%
7	280	93.33%
8	284	94.67%
9	283	94.33%
10	285	95.00%

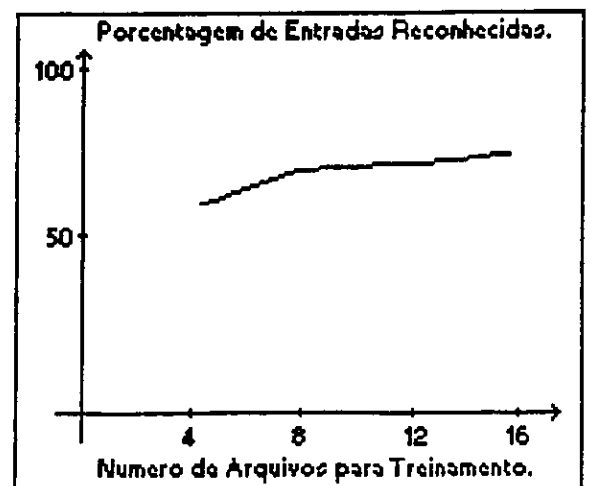


Fig. 5 - Gráfico da porcentagem de entradas reconhecidas x número de arquivos de 4 pessoas para treinamento.

O terceiro teste foi feito treinando-se a rede com 4 arquivos contendo cada um 10

números decimais de 4 pessoas distintas. Para o reconhecimento foram utilizados um total de 40 arquivos de pessoas diferentes daquelas usadas no treinamento, tentando fazer um sistema reconhecedor independente da pessoa. Neste caso a porcentagem de acerto caiu bastante, o seu valor foi de 57.75%. Em todos os testes anteriores usamos caracteres tendo uma dimensão de 30 x 30.

Quando usamos caracteres de dimensões 20 x 20 percebemos uma diminuição na porcentagem de acerto, como por exemplo a porcentagem de acerto foi de 88% quando usamos 5 arquivos de uma única pessoa para treinamento. Este valor foi de 90.33% quando se usou caracteres de dimensões 30 x 30.

Quando aumentamos a dimensões dos caracteres para 40 x 40, e fazendo-se o mesmo teste anterior a porcentagem de acerto passou para 93.00%.

Tabela 2 - Treinamento e reconhecimento com arquivos de 4 pessoas distintas das usadas no treinamto.

Número de Arquivos para Treinamento	Número de Entradas Reconhecidas	Porcentagem de Entradas Reconhecidas
4	225	56.25%
8	267	66.75%
12	276	69.00%
16	284	71.00%

**Conclusões:**

A rede funcionou satisfatoriamente quando foi utilizado os arquivos de uma única pessoa para treinamento e reconhecimento. Como era de se esperar quanto maior o número de arquivos utilizados para treinamento melhores foram os resultados. O mesmo ocorre quando se utiliza 4 pessoas distintas para treinamento. Podemos verificar que a porcentagem de acerto cai um pouco em relação a citada anteriormente, e concluímos que quanto maior o número de arquivos utilizados no

treinamento, maior a porcentagem de reconhecimento. Quando foram usados 40 arquivos de pessoas diferentes daquelas usadas no treinamento, a porcentagem cai bastante, mas alguns testes estão sendo feitos para melhorar os resultados. Na utilização de caracteres de dimensões 20 x 20 diminui-se a porcentagem de reconhecimento em relação a dos caracteres de dimensões 30 x 30 como era de se esperar, isso devido a menor quantidade de "pixels" utilizados no treinamento. Quando foi aumentada as dimensões dos caracteres para 40 x 40 podemos verificar que a porcentagem de acerto aumentou como era de se esperar, isto porque no treinamento foram utilizados mais "pixels" nas matrizes dos caracteres.

**Referências**

- 1 - C. Yüceer and K. Oflazer, A Rotation, Scaling, and Translation Invariant Pattern Classification System, Pattern Recognition 26 (5), pp. 687 - 710 (May 1993).
- 2 - H. A. Malki and A.Moghaddamjoo, Using the Karhunen-Loève transformations in the back-propagation training algorithm, IEEE Trans. Neural Networks 2 (1), pp. 162 - 165 (January 1991).
- 3 - M. Kirby and L. Sirovich, Application of the Karhunen-Loève procedures for the characterization of human faces, IEEE Trans. Pattern Analysis Mach. Intell. 12 (1), pp. 103 - 108 (January 1990).
- 4 - R. C. Gonzales and P. Wintz, Digital Image Processing, pp. 122 - 130. Addison - Wesley, Reading, Massachusetts (1987).
- 5 - D. J. Burr, Experiments on Neural Net Recognition of Spoken and Written Text, IEEE Trans. on Acoustics, Speech, and Signal Processing 36 (7), 1162 - 1168 (July 1988).