Mathware & Soft Computing 8 (2001) 291-301

A Neuro-Fuzzy System for Isolated Hand-Written Digit Recognition

M. Pinzolas[1], J.J. Astrain[2], J. Villadangos[3] and
J.R. González de Mendívil[2]
[1]Dpto. Ingeniería Sistemas y Automática. Univ. Politécnica Cartagena
Campus de la Muralla del Mar 30202 Cartagena.
e-mail: Miguel.Pinzolas@upct.es
[2]Dpto. Matemática e Informática. Univ. Pública de Navarra
Campus de Arrosadía 31006 Pamplona.
e-mail: josej.astrain,mendivil@unavarra.es
[3]Dpto. Automática y Computación. Univ. Pública de Navarra
Campus de Arrosadía,. 31006 Pamplona. *e-mail: jesusv@unavarra.es*

Abstract

A neuro-fuzzy system for isolated hand-written digit recognition using a similarity fuzzy measure is presented. The system is composed of two main blocks: a first block that normalizes the input and compares it with a set of fuzzy patterns, and a second block with a multilayer perceptron to perform a neuronal classification. The comparison with the fuzzy patterns is performed via a fuzzy similarity measure that uses the Yager parametric t-norms and t-conorms. Along this work, several values of the parameters have been studied, in order to obtain the best classification. The simplicity of the method makes it extremely quick and provides a recognition accuracy about 90% in classification of isolated digits, making it an attractive method for practical applications.

Keywords: hand-written digit recognition, neuronal networks, t-norms, t-conorms, fuzzy classification.

1 Introduction

Recognition of hand-written digits from scanned documents is a key step in document image processing. In the problem of optical character recognition (OCR) [6], the recognition of an isolated character or digit (ICR) is only a part of a complex problem that, in several cases, is recursive.

In these cases, the necessity of quick, reliable ICR method is critical. Most of the ICR methods proposed in the literature are based on two fundamental approaches [6, 7]: feature classification and template matching. The methods based on feature extraction need a complex, time-expensive treatment of the image that makes it difficult to use them in real time applications. On the other hand, template matching techniques are more sensitive to font and size variations of the characters, and hence they are less indicated for hand-written digit recognition [8]. Moreover, the variability of the hand-written digits makes the selection of desired templates a difficult problem. In order to deal with these drawbacks, the use of size normalization and fuzzy similarity measures is considered.

In the normalization process, the scanned digits are reduced to a fixed size by means of a zooming method [7]. The proposed method reduces the size of the digit by computing the density of pixels on the different zones of the image. These normalized images will be treated like fuzzy sets, which can be compared with the templates through a fuzzy similarity measure [5]. Thus, variability, noise and ambiguity in data can be considered thanks to the properties of fuzzy logic. In order to provide acceptable recognition rates, a neural network is used to classify the similarity values among the classes.

In this method, several parameters need to be fixed, such as the ones that define the similarity measure to be used, or the size of the normalization. This paper deals with the empirical study of the influence of these parameters on the performance of the method. To do this, the parametric t-norms and t-conorms of Yager [5] have been selected to perform the fuzzy comparisons, because they allows us to tune the strength of the fuzzy intersection and union. In the same way, various sizes for character normalization have been considered.

The paper is structured as follows: in the next section, the method is described. In section 3, the training of the system is explained. In section 4, results of the experiments performed are shown. Our conclusions about this work are summarized in section 5. Finally, an appendix with some examples of the data used in the training and validation phases is included.

2 The proposed method

The proposed ICR is composed by two main blocks as illustrated in figure 1.

The first block receives as input the digitized image of the digit. In this image, black pixels are attributed the value of one, and white ones are treated like zeros. The variability of the input image dimensions makes necessary to normalize the images. A zooming method is employed to reduce the original images in black and white to a $m \times m$ matrix. Each element of the matrix represents the pixel density in a zone of the original image,

$$M[i,j] = numberofpixelszone(i,j)/totalnumberofpixels. \tag{1}$$

This is a direct method, in opposition to more elaborated methods of feature extracting. In the resulting matrix, elements have a value between zero and one. This allows us to interpret it as a two-dimensional fuzzy set (figure 2).

$$\vec{x} = \{([i,j], M[i,j])\,;1 \leq i,j \leq m, M[i,j] \in [0,1]\} \tag{2}$$

Figure 1: Structure of the system

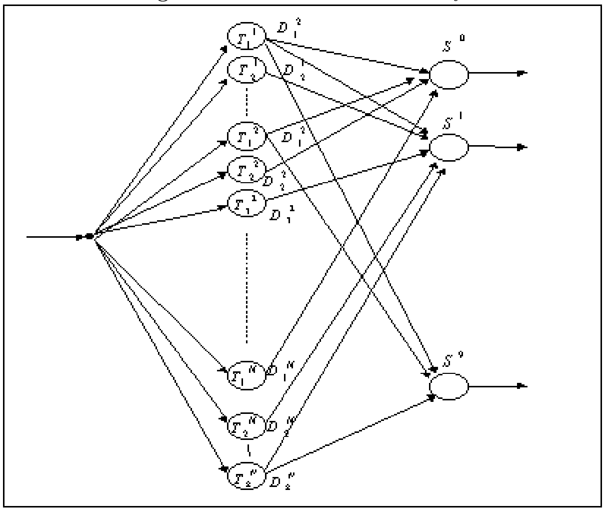
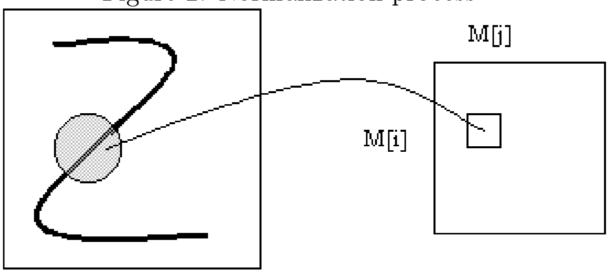


Figure 2: Normalization process



Due to the indicated variability, a set of templates for a given class is considered. The remaining digits in the class can be considered as noisy versions of this template. Dissimilarity measure between the patterns can be obtained, and this allows us to perform a fuzzy template matching.

Therefore, in a second step, a fuzzy comparison with the templates is performed, to provide the outputs of this first block. These outputs are interpreted as the dissimilarity fuzzy measures of the input image to the stored templates. To provide the similarity degree of two fuzzy sets A and B, the following method is used [5]:

$$E(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{3}$$

So, the dissimilarity measure is:

$$D(A,B) = 1 - E(A,B) \tag{4}$$

where $\cap$ and $\cup$ are the fuzzy union and intersection of A and B, respectively,

$|C|$ is the fuzzy cardinality of set C, defined as: $|C| = \sum_{x \in U} \mu_C(x)$

In this expression, $\mu_C(x)$ is the membership value of x in the fuzzy set C, being U the universe of discourse. On the other hand, fuzzy intersections and unions must be performed by means of t-norms and t-conorms, so that:

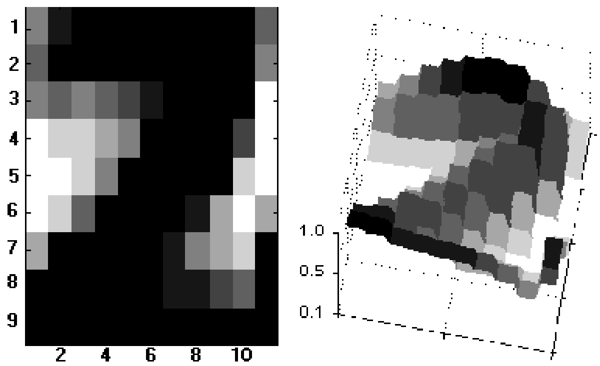
$$\begin{array}{l} \mu_{A\cap B}(x) = t\left[\mu_A(x), \mu_B(x)\right] \\ \mu_{A\cup B}(x) = s\left[\mu_A(x), \mu_B(x)\right] \end{array} \quad (5)$$

The definition of t-norms and t-conorms is not unique [2]. In this work, the parametric family of Yager t-norms and t-conorms has been used [10]:

$$\begin{array}{l} t_w^Y = 1 - \min\left(1, \left[(1-\mu_a)^w + (1-\mu_b)^w\right]^{\frac{1}{w}}\right) \\ s_w^Y = \min\left(1, \left[(1-\mu_a)^w + (1-\mu_b)^w\right]^{\frac{1}{w}}\right) \end{array} \quad (6)$$

The main feature of these norms is that as parameter w varies among 0 and infinite, the t-norms (and t-conorms) obtained span the space of fuzzy t-norms (and t-conorms) between the drastic product (drastic sum) and the minimum (maximum) [1], as shown in figure 3.

Figure 3: Yager parametric norms and co-norms



The outputs of the first block are the fuzzy dissimilarity values of the input pattern $\vec{x}$ to each of the fuzzy templates $\vec{T}_i^N$ (i-th template of the class N), that is, $D(\vec{x}, \vec{T}_i^N)$. At this point, it is possible to take as the final classification the digit class corresponding to the pattern that is closest to the input. However, a better approach is to consider that each of the fuzzy templates belongs to a different classifier. Thus, a multiple classification must be done to give the definitive output of the recognizer.

Among the several multiple classification methods [4], artificial neural networks have been selected, due to their ability of learning from examples and their reasonable interpolation behavior. Therefore, the second block of our system is a feed-forward multilayer perceptron [9]. The neural network is composed of a layer of ten sigmoid neurons. Each neuron is associated with one of the ten digit classes.

The inputs to these neurons are the fuzzy dissimilarity measures obtained in the first block. The output of each neuron, bounded between 0 and 1, can be interpreted as the possibility of the input image being the associated digit class, and it is given by the expression:

$$S^N = \frac{1}{\exp(-U^N)} \tag{7}$$

where U^N is the net input to the $N-th$ neuron in the layer, given by the expression:

$$U^N = \sum_{j=0}^{9} \sum_{i=0}^{k_j} D_i^j w_{ij}^N + b^N \tag{8}$$

where D_i^j is the dissimilarity between the input and the template $\vec{T}_i^j$, w_{ij}^N are the weights of the neuron S^N, and b^N is the bias.

The maximum is selected as the decision method; so that, the digit class associated to an input is the digit class having the greatest value in the output of the neural network.

3 Training system

For each class, the pattern template selection starts by choosing a group of kernels. For the first kernel $\vec{K}_1^N$ of class N a random sample is chosen among the examples belonging to the class. The second kernel, $\vec{K}_2^N$, is the sample of the same class having the maximum dissimilarity with respect to the previous kernel $\vec{K}_1^N$. The third kernel ($\vec{K}_3^N$) is the sample having the maximum of the added dissimilarities with respect to the first and second kernels, and so on.

Formally, let $samples(N)$ be the set of samples of the class N and

$$\left\{\vec{K}_i^N, \vec{K}_i^N \in samples(N), i : 1 \ldots p-1\right\} \tag{9}$$

the set of the chosen $p-1$ first kernels to the class N.

The new template p, $\vec{K}_i^N \in samples(N)$, is the kernel that verifies:

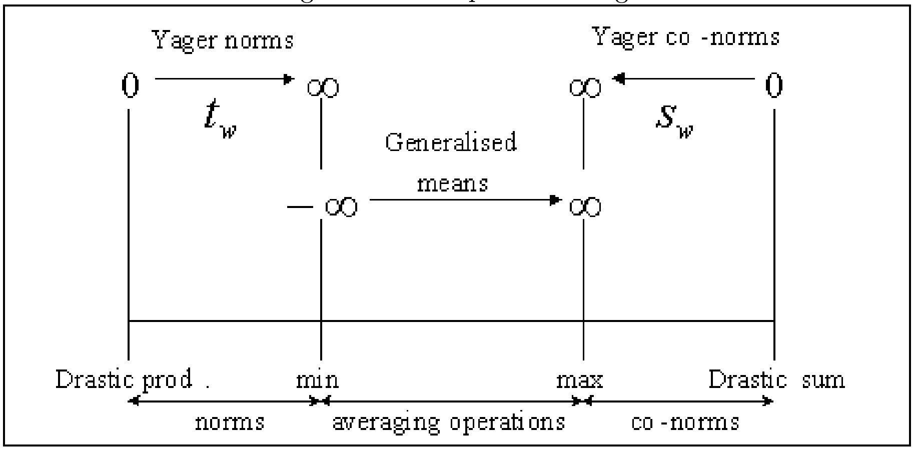
$$\forall \vec{x}_r^N \in samples(N) : \sum_{i=1}^{p-1} D\left(\vec{K}_p^N, \vec{K}_i^N\right) \geq \sum_{i=1}^{p-1} D\left(\vec{x}_r^N, \vec{K}_i^N\right) \tag{10}$$

Once the kernels of every class have been selected, the samples will be divided in as many groups as kernels, using a proximity criterion to each kernel. Each sample is associated to the group corresponding to the closest kernel. Once all the samples have been associated to the respective groups, the pattern template $\vec{T}_i^N$ of each group is chosen as the fuzzy set resulting of averaging all the samples in the group. More formally, let $\left\{\vec{K}_i^N, \vec{K}_i^N \in samples(N), i : 1 \ldots p-1\right\}$ be the set of the p templates chosen for class N; and $C(\vec{K}_i^N) \subset samples(N)$ the union of samples of the class N; initially $C(\vec{K}_i^N) = \{\vec{K}_i^N\}$. A sample $\vec{x}_r^N \in sample(N)$ is associated to a $C(\vec{K}_i^N)$, $\vec{x}_r^N \in C(\vec{K}_i^N)$, if and only if:

$$\forall j : 1...p, D\left(\vec{x}_r^N, \vec{K}_i^N\right) \leq D\left(\vec{x}_r^N, \vec{K}_j^N\right) \tag{11}$$

An example of a $10x10$ pattern for digit '2' is shown in figure 4.

Figure 4: 10x10 pattern of digit '2'



Finally, the $i-th$ pattern template of the class N is the pattern $\vec{T}_i^N = < C(\vec{K}_i^N) >$.

Once the pattern templates have been fixed, it is necessary to train the neural network. A key decision when using neural networks is the selection of the learning method. In this work, the Levenberg-Mardquardt [3] learning method was selected, because of his good performance when the number of trainable weights is not very large. Along this work, more complicated network structures involving a hidden layer were considered. Although the error reached in the learning phase is better than in the one layer case, the behavior in the validation is worst, due to over-fitting.

4 Experimental results

The aim of the experiments was to test the reliability of the method, and to adjust parameters such as size of normalization, number of templates and fuzzy dissimilarity measure, which have influence on its performance.

In order to test the sensitivity of the method with regard to the size of normalization and to the number of templates, several tests were performed. Four sizes of normalization were considered: $6x6$, $8x8$, $10x10$ and $16x16$. For each size of normalization, several values of the Yager parameter were considered, in an effort to tune the dissimilarity measure.

	w=0	w=1	w=2	w=3	w=4	w=5	w=10	w=20	w=10^3	w=∞
4 Temp.	23.27	80.13	91.05	92.48	93.27	93.40	93.40	93.07	90.89	92.94
3 Temp.	23.27	77.19	90.20	90.78	91.90	92.42	92.61	91.44	88.69	91.50
2 Temp.	23.27	76.67	87.25	88.95	89.35	89.61	89.02	88.50	84.58	87.84
1 Temp.	20.65	71.57	83.73	83.20	83.79	84.18	84.77	84.31	78.24	83.92

Table 1: Results after training, 6x6 normalization

	w=0	w=1	w=2	w=3	w=4	w=5	w=10	w=20	w=10^3	w=∞
4 Temp.	8.10	77.53	87.33	88.09	89.75	89.29	88.96	88.57	87.78	88.70
3 Temp.	8.10	74.72	87.72	87.85	88.70	88.37	88.90	88.11	86.35	88.05
2 Temp.	8.10	72.89	84.42	86.54	86.61	86.68	85.83	85.63	82.04	85.43
1 Temp.	16.33	70.08	83.08	82.82	83.15	83.21	82.69	82.17	76.16	81.71

Table 2: Results after validation, 6x6 normalization

The digits used in the experiments were scanned from the courtesy amount of real cheeks from Hispano-America and Europe.

The samples were divided into two separated sets of 1530 samples, one for training purposes and the other for validation (see Appendix). In the following tables, results of experiments are shown. Tables 1, 3, 5, and 7 show the percentage of success obtained after training, with the first set of examples. In tables 2, 4, 6, and 8 the percentage of success for the validation set can be seen.

Results show that the best performance is reached when normalizing digits at $6x6$ and the Yager t-norm and t-conorm with parameter $w = 4$ are used. It can be seen that the smallest normalization size gives the best results. This indicates that normalization process eliminates spurious data that could confuse the classifier. The worst results are obtained when the drastic sum and product are used and normalization size is $6x6$. This is because drastic sum and product are very insensitive to variations on the input patterns, and $6x6$ normalization makes variations be smaller.

The small difference between the training and validation results points to a good generalization has been achieved, due to a correct dimensioning of the network employed and to a sufficiently representative set of training examples.

Some additional experiments were performed in order to test the influence of using a greater number of templates, showing that no significant improvements on the percentage of recognition are obtained, although the number of templates increases.

	w=0	w=1	w=2	w=3	w=4	w=5	w=10	w=20	w=10^3	w=∞
4 Temp.	37.78	88.82	92.94	93.14	93.40	93.79	92.61	93.14	92.61	93.27
3 Temp.	37.78	86.41	90.39	91.63	92.48	91.90	91.31	91.44	90.00	91.24
2 Temp.	37.78	83.79	87.97	87.45	88.63	89.28	88.04	88.37	86.99	88.30
1 Temp.	34.25	80.00	83.33	83.53	83.86	83.66	82.61	82.48	80.98	82.42

Table 3: Results after training, 8x8 normalization

	w=0	w=1	w=2	w=3	w=4	w=5	w=10	w=20	w=10^3	w=∞
4 Temp.	33.31	82.76	88.18	88.83	88.90	88.70	88.05	88.31	87.85	87.98
3 Temp.	33.31	81.37	87.26	88.37	87.72	87.72	86.54	87.39	85.96	87.13
2 Temp.	33.31	80.34	85.30	86.02	86.15	85.96	85.50	84.91	82.23	84.91
1 Temp.	31.55	78.05	82.63	82.76	82.63	82.69	81.91	81.32	77.34	81.12

Table 4: Results after validation, 8x8 normalization

	w=0	w=1	w=2	w=3	w=4	w=5	w=10	w=20	w=10^3	w=∞
4 Temp.	54.51	91.80	93.30	93.60	93.90	94.20	93.80	93.30	91.76	93.46
3 Temp.	54.51	88.30	91.50	93.60	92.20	92.20	91.40	91.50	89.74	91.63
2 Temp.	54.51	85.80	89.30	88.40	88.80	89.00	88.60	88.80	86.93	89.08
1 Temp.	51.50	80.10	83.70	83.80	83.90	83.90	83.00	82.80	79.61	82.75

Table 5: Results after training, 10x10 normalization

	w=0	w=1	w=2	w=3	w=4	w=5	w=10	w=20	w=10^3	w=∞
4 Temp.	51.86	87.00	88.60	88.80	89.20	89.00	88.20	88.70	87.00	88.05
3 Temp.	51.86	84.40	87.70	88.20	88.40	88.40	87.60	88.70	85.70	87.63
2 Temp.	51.86	83.50	85.80	86.40	86.20	86.20	85.80	85.20	82.10	85.04
1 Temp.	50.69	79.60	83.00	83.20	83.40	83.20	82.20	81.80	76.03	81.91

Table 6: Results after validation, 10x10 normalization

	w=0	w=1	w=2	w=3	w=4	w=5	w=10	w=20	w=10^3	w=∞
4 Temp.	82.55	92.68	93.46	93.27	93.86	93.53	93.33	92.94	93.33	92.94
3 Temp.	82.29	91.24	91.24	91.44	91.63	91.70	91.11	90.78	91.83	90.92
2 Temp.	80.46	88.17	88.63	88.69	88.04	88.30	88.37	88.24	89.22	88.10
1 Temp.	77.71	83.20	84.58	84.31	84.31	83.99	84.05	84.05	82.88	83.79

Table 7: Results after training, 16x16 normalization

	w=0	w=1	w=2	w=3	w=4	w=5	w=10	w=20	w=10^3	w=∞
4 Temp.	77.92	87.39	89.09	88.24	88.90	88.31	88.37	88.57	88.31	88.44
3 Temp.	77.20	86.61	87.72	87.72	87.79	87.07	87.26	87.20	86.61	87.20
2 Temp.	77.66	84.78	85.56	85.11	85.89	85.63	85.43	85.11	85.11	85.30
1 Temp.	74.40	82.56	82.56	82.89	82.43	82.36	82.23	81.97	79.62	81.97

Table 8: Results after validation, 16x16 normalization

5 Conclusions

A method to recognize isolated hand-written digits making use of fuzzy dissimilarity and multiple classification based in neural networks is proposed. The experiments performed show that the performance of the method depends on various parameters, mainly the number of templates employed, size of the normalized digits and fuzzy norms and co-norms selected. In this work, an empirical tuning of these parameters has been carried out. Results show that it is not necessary to employ a high number of templates for the matching because it does not increase the percentage of recognition obtained. Another result observed is that using a $6x6$ normalization, an acceptable recognition rate close to the 90% in the validation phase, and 93% in training phase, is obtained. The simplicity of the method offers a low computational cost when compared with other methods mentioned in the literature, while maintaining similar recognition accuracy.

6 Acknowledgments

The authors wish to thank to the enterprise Investigación y Programas, S.A. (IPSA) that has contributed with all the images to build the database of digits, also with financial support.

7 Appendix

In order to train the classifier, a set of images coming from Investigación y Programas, S.A. (IPSA) enterprise has been employed. All the samples used in this work have been extracted from numeric hand-written amounts of real cheques. The segmentation of isolated digits has been carried out to construct a database. The cheques, originating from Mexico, Argentina and Colombia, are very different among them.

The number of samples per digit in the training database is proportional to the appearance of each digit in the set of samples analysed. The validation database presents a uniform number of samples distribution.

A series of digits coming from the validation database is shown in figure 5

Figure 5: Some of the digits used for validation

References

[1] Chin-Teng Lin. C.S. George Lee. Neural Fuzzy Systems, Ed. Prentice-Hall, 1996.

[2] G. J. Klir, B. Yuan. Fuzzy sets and fuzzy logic. Prentice-Hall, 1995.

[3] G.Lera, M. Pinzolas. A quasi-local Levenberg-Mardquardt algorithm for neural network training. IJCNN'98 IEEE World Congress on Computation Intelligence, Proceedings of the IJCNN'98, pp. 2242-2246, 1998.

[4] Y. Lu, F. Yamaoka. Fuzzy integration of classification results. Pattern Recognition, Vol. 30, No 11, pp. 1877-1891, 1997.

[5] Pedrycz, Fuzzy Control and Fuzzy Systems. Ed. Research Studies Press Ltd, 1993.

[6] Shunjimori et als. Historical review of OCR research and development. Proceedings of the IEEE, vol. 80, No 7, July 1992.

[7] T. Steinherz, E. Rivlin, N. Intrator, Off-line cursive script word recognition-a survey. International Journal on Document Analysis and Recognition, 1999.

[8] S. Sural, P.K.Das, Fuzzy Hough transform and an MLP with fuzzy input/ output for character recognition. Fuzzy Sets and Systems, 105 pp.489-497, 1998.

[9] B. Widrow, M.A. Lehr. 30 years of adaptive neural networks: Perceptron, Madaline and backpropagation. Proceedings of the IEEE, Vol. 78, No 9, pp. 1415-1442, 1990.

[10] R.R. Yager. On a general class of fuzzy connectives. Fuzzy Sets and Systems, 4: 235-242, 1980.