Neural Network for Numeral Recognition

Soukaina Benchaou
Labo MATSI, EST
University Mohammed 1
Oujda, Morocco

M'Barek Nasri
Labo MATSI, EST
University Mohammed 1
Oujda, Morocco

Ouafae El Melhaoui
LaboMATSI, EST
University Mohammed 1
Oujda, Morocco

ABSTRACT

The present paper proposes a new approach of preprocessing for handwritten, printed and isolated numeral characters. The new approach reduces the size of the input image of each numeral by discarding the redundant information. This method reduces also the number of features of the attribute vector provided by the extraction features method. Numeral recognition is carried out in this work through k nearest neighbors and multilayer perceptron techniques. The simulations have obtained a good rate of recognition in fewer running time.

General Terms

Pattern Recognition, image processing, feature extraction, neural network.

Keywords

Handwritten and printed numeral recognition, preprocessing, profile projection, k nearest neighbors, multilayer perceptron.

1. INTRODUCTION

For the last decade, the recognition of handwritten characters has known a big progress in numerous industrial applications, in particular the numeral recognition which is used in several sectors such as postal sorting, bank check reading, order form processing, etc.

The recognition system requires three main steps to predict the class of membership of an unknown pattern, which are: preprocessing, features extraction and classification.

The preprocessing phase consists of discarding the imperfections and reducing the analyzed area.

The Numeral features extraction is a delicate process and is crucial [1] for a good numeral recognition. It consists of transforming the image into an attribute vector which contains a set of discriminated characteristics for recognition and also reducing the amount of information supplied to the system.

In the literature, several works have been proposed for features extractions such as invariant moments [2], Zernike moments [3], freeman coding [4], Loci characteristics [5], etc.

The last step is the classification which consists of partitioning a set of data entity into separate classes according to a similarity criterion, different methods are proposed in this context includesdynamic programming, neural networks, support vector machines, k nearest neighbors, k-means, etc.

In order to validate our contributions, we have used in this work a database of 590 numerals,printed andhandwritten, provided by various categories of writers.

In section 2 numeral features extraction method is presented. The k nearest neighbors and the multilayer perceptron techniques of classification are discussed in section 3. The proposed system for numeral recognition is presented in section 4. The result of simulations and comparisons are introduced in section 5. Finally, we give a conclusion.

2. NUMERAL FEATURES EXTRACTION

Features extraction is a complex task. It knows a great interest in multiple domains such as image processing, segmentation, classification, etc.

After the preprocessing step, the preprocessed image is represented by a matrix of pixels which can be of very large size. So, it will be useful to represent objects by characteristics containing the necessary information. This operation is called features extraction. In the literature, the feature extraction methods are classified according to two categories: statistical approach and structural approach. In our work we are going to use the statistical approach based on profile projection.

2.1 Profile Projection

Profile projection is a statistical method successfully used in recognition of handwritten characters [6], etc.
The preprocessing stage consists of binarizing the numeral input image which is presented in grey level. The next stage is to only preserve the numeral position in image by cropping it. Final stage is to fix the size of cropped numeral image.

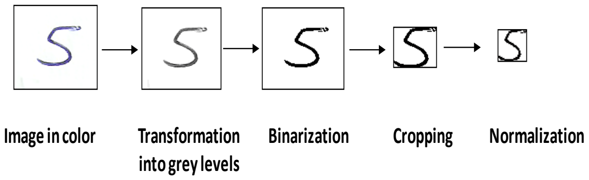

Fig 1: Different steps for preprocessing of numeral '5'

This method calculates the number of pixels (distance) between the left, bottom, right, top edge of the image and the first black pixel met on this row or column. The dimension of the obtainedattribute vector is twice the sum of the number of rows and columns associated to the image of the numeral.

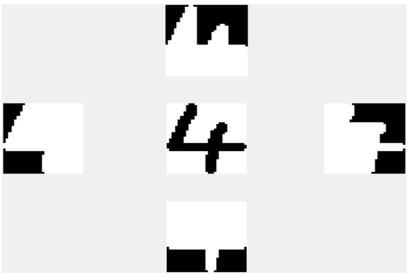

Fig 2: The four profile projections of numeral '4'

3. CLASSIFICATION METHODS

3.1 K nearest neighbors

K nearest neighbors (KNN) is a widely used method for data classification. Proposed in 1967 by Cover [7], it has been widely used in handwritten numerals recognition for its simplicity and its robustness [8].

KNN is a method which was inspired from the closest neighbor rule. It is based on computing the distance between the test sample and the different learning data samples and then attributes the sample to the k nearest neighbors.

```
Algorithm 1 : K nearest neighbors

- Let L = {(x′, c)/x′ Є, $\mathbb{R}^N$, c = 1,2,…, □} the learning set.
- Let x the example to determine its class.
  1 : **For each** object x′ of the set L
 - Compute the distance between the object x and the
       object x′, d(x, x′).
- Classify the different distances in increasing way.
**End**
 2 : **For every** {x’ Є KNN(x)} **do** :
- Identify the most frequent class.
**End**
3 : Assign to x the identified class.
```

3.2 Multilayer perceptron

Neural networks are one branch of artificial intelligence. Artificial neural network is defined as a computer system who has been inspired from a long study of human brain and how the human neurons function [9].

Artificial neural networks in general and multilayer perceptron (MLP) in particular have been widely used for data classification, pattern recognition (characters, voice, signal, etc.) [10][11], prediction [12], etc. The MLP is characterized by its ability to learn and gradually improve its performance through a learning process.

MLPs are forward propagation networks where the two closest layers are fully connected. The MLP structure contains an input, an output and a certain number of hidden layers.

We have been limited, in this work to one hidden layer.

Learning is a phase where the behavior of the network is modified by modifying the synaptic weights until a desired output pattern is obtained.

In this work, we have used the gradient backpropagation algorithm, the objective is to minimize the squared error between the desired and computed output of the MLP. Algorithm 2 shows the different steps of the gradient backpropagation algorithm.

```
Algorithm 2: Gradient backpropagation

1. Randomly initialize the synaptic weights between -1 and 1.
2. Randomly apply a realization vector of an object to the input layer and its corresponding known output to the output layer.
3. Compute the network output and error E between computed and desired outputs.
4. Adjust the weights by the gradient method:
      $W^{(r)}(t+1) = W(t)^{(r)} - \eta \frac{\partial E}{\partial W^{(r)}}$
   η is the learning rate, which is in general a value between 0.1 and 0.9, r = 1,2.
5. Go to 2 as long as the network does not show satisfactory performances.
```

We have a structure of three layers:

- The input layer has n1 neurons that we call nek, $1 \le k \le n_1$,
- The hidden and output layers contain n_2 and n_3 neurons that we respectively call ncj and nsi, where $1 \le j \le n_2$, $1 \le i \le n_3$.

$z_k^{(1)}$ is the realization of the attribute vector component for a numeral image I, with k=1,2,⋯,n_1.

$W^{(1)} = (w_{jk}^{(1)})$, j=1, ⋯,n_2, k=1, ⋯,n_1, $w_{jk}^{(1)}$ are the synaptic weights connecting neurons in the input to the neurons in the hidden layers.

$W^{(2)} = (w_{ij}^{(2)})$, i=1,⋯,n_3, j=1, ⋯,n_2, $w_{ij}^{(2)}$ are the synaptic weights connecting neurons in the hidden to the neurons in the output layers. The output of the neuron j, ncj of the hidden layer is:

$$z_j^{(2)} = f(y_j^{(2)}) \tag{1}$$

Where
$$y_j^{(2)} = \sum_{k=1}^{n_1} w_{jk}^{(1)} z_k^{(1)} \tag{2}$$

And
$$f(y_j^{(2)}) = \frac{1}{1+e^{-y_j^{(2)}}} \tag{3}$$

For j=1,2,⋯,n_2, f is the activation function which we choose to be of sigmoid type.

The output of the neuron i , ns_i of the output layer is:

$$z_i^{(3)} = f(y_i^{(3)}) \tag{4}$$

Where
$$y_i^{(3)} = \sum_{j=1}^{n_2} w_{ij}^{(2)} z_j^{(2)} \tag{5}$$

For i= 1,2,⋯,n_3.

Notice that the superscripts (1), (2) and (3) represent respectively input, hidden and output layers.

- Optimization of the $w_{ij}^{(2)}$:

In order to be able to update $w_{ij}^{(2)}$ synaptic weights for different indices i and j, consider the network error to be:

$$E(t) = \frac{1}{2} \sum_{i=1}^{n3} (z_i^{(3)} - d_i(t))^2 \tag{6}$$

Where $d_i(t)$ is the known desired output at neuron nsi. t represents the current iteration. Differentiating $E(t)$ with respect to $w_{ij}^{(2)}$ gives:

$$\frac{\partial E(t)}{\partial w_{ij}^{(2)}} = (z_i^{(3)}(t) - d_i(t)) \frac{\partial z_i^{(3)}}{\partial w_{ij}^{(2)}} \tag{7}$$

Where
$$\frac{\partial z_i^{(3)}}{\partial w_{ij}^{(2)}} = z_j^{(2)} \times f(y_i^{(3)}) \times (1 - f(y_i^{(3)})) \tag{8}$$

The synaptic weight $w_{ij}^{(2)}$ may now be updated.

$$w_{ij}^{(2)}(t+1) = w_{ij}^{(2)}(t) - hd_i^{(3)}(t) z_j^{(2)}(t) \quad (9)$$

For i=1,⋯,n$_3$ and j =1,⋯,n$_2$
Where

$$d_i^{(3)}(t) = (z_i^{(3)}(t) - d_i(t)) f(y_i^{(3)})\text{´}\ (1 - f(y_i^{(3)})) \quad (10)$$

- Optimization of the $w_{jk}^{(1)}$

The derivation in this case is similar to that presented previously, so we go to the updating of the weights according to:

$$w_{jk}^{(1)}(t+1) = w_{jk}^{(1)}(t) - hd_j^{(2)}(t) z_k^{(1)}(t) \qquad (11)$$

For j =1,⋯,n$_2$ and k=1,⋯,n$_1$
Where

$$d_j^{(2)}(t) = \overset{n_3}{\underset{i=1}{\text{å}}} d_i^{(3)}(t) w_{ij}^{(2)}(t) \frac{\P f(y_j^{(2)}(t))}{\P y_j^{(2)}(t)} \qquad (12)$$

4. PROPOSED SYSTEM

We have used a database of 590 Arabic numerals, provided by different writers. A sample of the database is shown in figure 3. The database is divided into two sets, one set of 400 numerals is used for learning and the remaining 190 numerals are used for the test stage.

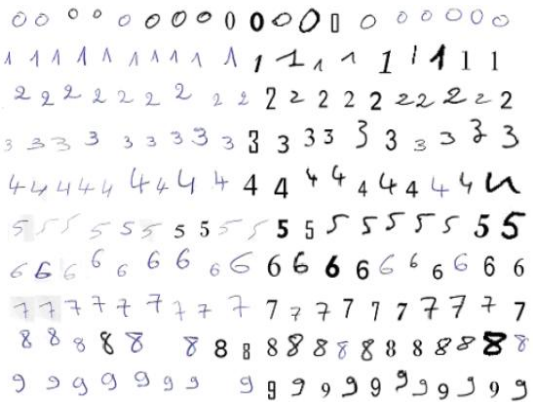

Fig 3: A sample of handwritten and printed numerals

The proposed system consists of adding a step of windowing in the preprocessing stage, in order to reduce the size of the attribute vector, which decreases the running time.

The proposed system contains three main steps, preprocessing, features extraction and classification. The full system is shown in figure 4.

The preprocessing stage, in our case, is so fast in computing time. It consists of binarizing the numeral input image which is presented in grey level. Then, wepreserve only the numeral position in image by cropping it. The next stage is to normalize the image in a predefined size. Finally, we have applied the windowing stage. This operation consists of dividing the numeral image into small windows. For every window we calculate the average of pixel grey level. This operation reduces widely the size of the image and preserves the useful information for recognition of every input image. So, the dimension of attribute vectors applied to the system will be reduced too. The three last steps (cropping, normalization andwindowing) reduce also widely the computing time of processing.

After preprocessing, features extraction is carried out by profile projection method. KNNand MLP methods are used for classification task.

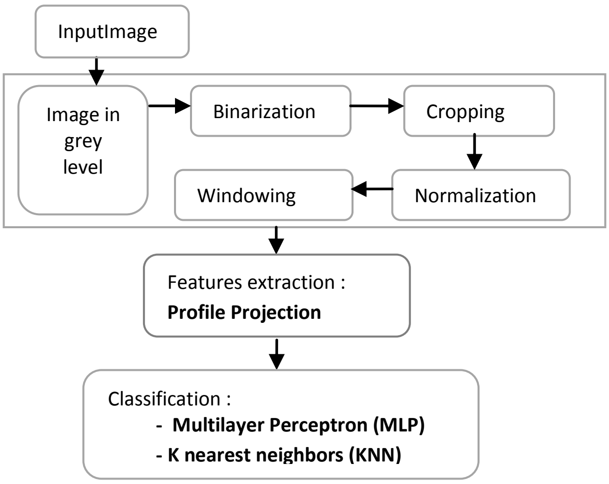

Fig 4: Scheme for numeral recognition

5. EXPERIMENTAL RESULTS AND COMPARATIVE STUDY

In the first test, the simulations of numeral recognition are carried out without using windowing stage in the preprocessing phase. Several experiments were carried out to determine the recognition rate according to the size of normalization and the number of neurons in the hidden layer of the multilayer perceptron.

Table 1 shows that the couple son and nc have an effect on recognition rate. In fact, with the couple (son,nc)= (40x40,40) we have achieved the maximum recognition rate equal to 86.8%. The attribute vector obtained contains 160 elements for each numeral.

Table 1: Numeral recognition rate according to the size of normalization and the number of neurons in the hidden layer

Size of normaliza-tion(son)	Profile projection		
	Number of primitives	Number of neurons in the hidden layer (nc)	Recognition rate %
20x20	80	15	77.89
		20	**82**
		30	81.05
30x30	120	25	82.63
		35	**84**
		45	83.68
40x40	160	30	78.9
		40	**86.8**
		55	84
50x50	200	50	84.73
		80	**85.26**
		140	83.16

Figure 5 illustrates the evolution of recognition rate according to the number of neurons in the hidden layer, using the size 40x40 as size of normalization.

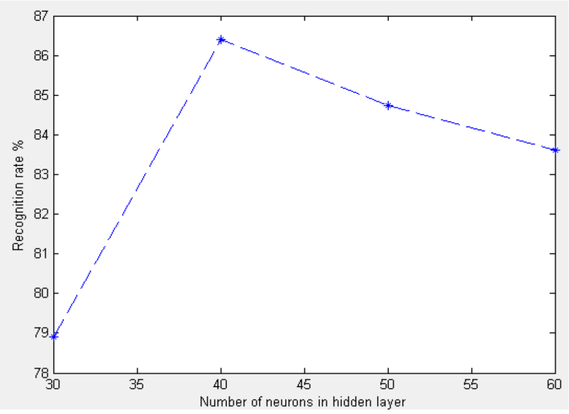

Fig 5: Evolution of recognition rate according to the number of neurons in the hidden layer

A normalization size of 40x40 obtains good recognition rate performance of isolated numerals. This size will be preserved for application of our proposed system.

In the classifier, the number of neurons in the input layer is too high. It is equal to 160 which weakened the performance of the multilayer perceptron and increased the time of processing.

For this reason, and in order to reduce the number of input neurons of the numeral recognition system, we have proposed the windowing step which consists to split the numeral image into equal areas of m pixels. For each area, the average of pixel grey level is calculated. This operation reduces the size of input image, attribute vector dimension, error rate and time processing.

In the second test, we have fixed the size of normalization on 40x40. Then, we have applied our approach of windowing in preprocessing phase.

Table 2 illustrates the numeral recognition rate using the proposed system after several simulations according to the number of hidden neurons.

Table 2: Numeral recognition rate of proposed system

Size of windowing	Number of primitives	Multilayer perceptron		K nearest neighbors	
		Recognition rate (%)	Time processing (s)	Recognition rate (%)	Time processing (s)
Without windowing	160	86.8	13.58	86.8	31
Window of 2 pixels	120	86.8	13.5	86.8	24
Window of 4 pixels	80	**87.9**	**4**	**87.9**	**19**

Table 2 highlights the fact that with the operation of windowing, the size of the attribute vector is reduced to 120 and 80 while it was initially 160. The time processing using the K nearest neighbors and multilayer perceptron is reduced too. We also notice that the choice of the size of windows is essential. The window of twopixels preserves the same recognition rate. It is equal to 86.8%. But the window of four pixels improves the recognition rate to 87.9%, this justifies that the redundant attributes are partially eliminated.

6. CONCLUSION

In this work we have presented a new approach for numerals recognition. In the preprocessing stage we have added the step of windowing. It consists to split the numeral image into equal areas of m pixels. For each area, the average of pixels is calculated. Features extraction is carried out byprofile projection. Multilayer perceptron method and K nearest neighbors are used for classification. The simulations have obtained good results; a clear reduction of attributes vector size and time processing with a good increase of recognition rate are obtained. That confirms the performance of our proposed system for numeral recognition.

7. REFERENCES

[1] Benne, R.G.1.,Dhandra, B.V.1. and Hangarge, M., 2009.Tri-scripts handwritten numeral recognition: a novel approach. Advances in Computational Research, ISSN: 0975–3273, Vol 1, Issue 2, pp: 47-51.

[2] Anuja, P. July 2011.License Plate Character Recognition System using Neural Network. International Journal of Computer Applications (0975-8887), Vol.25, N°.10.

[3] Abu Bakar,N., Shamsuddin,S., and Ali,A. 2010. An integrated formulation of zernike representation in character images. Trends in Applied Intelligent Systems, volume 6098 of Lecture Notes in Computer Science, pages 359–368. Springer Berlin / Heidelberg.

[4] Dupré,X. 2003. Contributions à la reconnaissance de l'écriture cursive à l'aide de modèles de Markov. Thèse, Université René Descartes-Paris 5.

[5] Ebrahim,R., Moradian, M.R., Esmkhani, A., Jafarlou,F.M. 2009. Recognition of Persian handwritten digits using characterization Loci and mixture of experts. International Journal of Digital Content.Technology and its Applications, Vol.3,N°.3.

[6] Jean_luc, H. 2008. Une étude sur le choix des caractéristiques pour la représentation de caractères imprimés. Vision Interface '99, Trois-Rivières, Canada.

[7] GIL-PITA,R., YAO,X. 2008. Evolving edited k-nearest neighbor classifiers. International Journal of Neural Systems, Vol.18, N°.6, pp 459–467.

[8] Bernard, M., Fromont,E., Habard, A., Sebban, M. 2012. HandWritten Digit Recognition using Edit Distance based KNN.

[9] Adnan,A. and Sameer,S. 1998. Optical character recognition: Neural network analysis of hand-printed characters. Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition, London, UK, Springer-Verlag.

[10] Pinto,J.P. 2010. Multilayer perceptron based hierarchical acoustic modeling for automatic speech recognition. Thèse, Ecole Polytechnique fédérale de Lausanne, Suisse.

[11] Oukhellou,L. 1997. Paramètrisation et classification de signaux en contrôle non destructif. Application à la reconnaissance des défauts de rails par courant de foucault. Thèse, Université Paris XI Orsay.

[12] Bouziane, H., Messabih, B., Chouarfia, A. 2008. Prédiction de la structure 2D des protéines par les réseaux de neurones. Communication of IBIMA, Vol.6.