GEOMETRIC SHAPE RECOGNITION USING FUZZY AND NEURAL TECHNIQUES

Ioan Z. MIHU*, Arpad GELLERT*, Cosmin N. SUCIU**

** "Lucian Blaga" University of Sibiu, Computer Science Department, str. Emil Cioran, nr. 4, Sibiu, ROMANIA, E-mail: ioan.z.mihu@ulbsibiu.ro, arpad.gellert@ulbsibiu.ro*

*** "S.C. Polisano S.R.L.", str. Moldoveanu, nr. 25, Sibiu, ROMANIA, E-mail: suciu_nicu@yahoo.com*

Abstract: This paper is focused on on-line geometric shape recognition (Ulgen et al. 1999) based on fuzzy techniques and backpropagation neural algorithm. We propose a new method for geometric shape recognition using a fuzzy classifier of angles and a multilayer neural network for training and classification of geometric shapes. Our method examines the geometric shape as a whole in a way similar to human recognition process. We have to use in the recognition process information that is invariant in terms of scaling, translation and rotation. The internal angles represent the relevant information relatively to the geometric shape. The key concept is that the neural network learns the internal angles of a shape.

Keywords: Geometric Shape Recognition, Neural Networks, Fuzzy Techniques

1. INTRODUCTION

The artificial neural networks are composed of a multitude of neurons, simple processing elements that operates in parallel. A great advantage of the artificial neural networks is their capacity to learn on exemples. To solve a problem traditionally, we have to elaborate its model, and after that we have to indicate a succession of operations that represents the solving algorithm of the problem. However there are practical problems with a high level complexity, and for this kind of problems is very hard or even impossible to establish an algorithm.

In the conexional models we are not forced to give a solving algorithm of a problem to the neural network, we have to offer him only a multitude of consistent exemples. The network extracts the information from the training samples, in this way it is able to synthetize implicitly a certain model of the problem. In other words, the neural network builds up alone an algorithm to solve a problem.

The capacity of the neural network to solve complex practical problems using a multitude of samples gives them a highly large potential of aplicability. In this work we use a multilayer feedforward neural network to recognize the basic geometric shapes.

2. THE MULTILAYER FEEDFORWARD NEURAL NETWORK

Multilayer neural networks are used for pattern classification, pattern matching, and function approximation. By adding a continuously differentiable function, such as Gaussian or sigmoid function, it is possible for the network to learn practically any nonlinear mapping to any desired degree of accuracy. There are several ways that multilayer neural networks can have their connection weights adjusted to learn mappings. The most popular technique is the backpropagation algorithm and its many variants.

Multilayer networks make it possible to implement any arbitrary function $\vec{y} = \Phi(\vec{x})$, $\vec{x}$ being the input of the first layer and $\vec{y} = \vec{y}^{[L]}$ representing the output of the last layer L. Often, the activation function σ is a hyperbolic tangent. The function Φ is learned by repeated presentation of input-output pairs $\{\vec{x}, \vec{d}\}$, called prototypes. The backpropagation (BP) learning rule is a gradient-descent algorithm that updates the weights to minimize the square-error on the learning prototypes. For that purpose an error signal is computed for each layer (Zurada 1992):

$$\delta_i^{[L]} = \left(d_i - y_i^{[L]}\right) \cdot \sigma'\left(p_i^{[L]}\right)$$

$$\delta_i^{[q]} = \left(\sum_{k=1}^{m_{q+1}} W_{k,i}^{[q+1]} \cdot \delta_k^{[q+1]}\right) \cdot \sigma'\left(p_i^{[q]}\right)$$

for q = 1, 2, …, L-1;

where $\sigma'(v) = \dfrac{d\sigma(v)}{dv}$. The equations are valid for all neurons i = 1, 2, …, m_q of layer q. Once the errors have been back-propagated, the weights are updated as:

$$W^{[q]} = W^{[q]} + \alpha \cdot \vec{\delta}^{[q]} \cdot \vec{y}^{[q-1]T}$$

for q = 1, 2, …, L;

where $\vec{y}^{[0]} = \vec{x}$.

3. GEOMETRIC SHAPE RECOGNITION

The classical techniques based on shape partitioning into segments, followed by a syntactical analysis to match with a predefined shape, are strongly affected by noise and are weak in terms of generalization. In order to eliminate these limitations of the classical methods, our method examines the geometric shape as a whole in a way similar to human recognition process. Human beings recognize such basic shapes regardless of the variations in size, noise on the shape border, translation, rotation, and in the case of triangles, regardless of the type of the triangle. That means that not the segments are important in the recognition process but the angles, which represents the relevant information relatively to the geometric shape. The key concept is that the neural network learns the internal angles of a shape (the angles between any two consecutive tangent vectors). As a consequence, the neural network training process will be simplified, therefore only a few training samples that represent a class of shapes (i.e. triangles, rectangles and elliptic shapes) are sufficient. Our application's aim is to recognize the basic geometric shapes (elliptic, rectangular and triangular).

3.1. Feature Extraction

The purpose of preprocessing is to create an intermediate representation of the input data and is performed on-line (prior to the application of recognition task). The preprocessing step can be defined as a feature extraction process that is important since it prepares input data that is invariant in terms of scaling, translation and rotation. The feature extraction is performed on the captured points along the boundary of the shape.

Since the geometric shapes are hand-drawn using the mouse, the information could include noise due to the variations in capture speed of the mouse and erratic hand motion while drawing. We have to extract the features of the shape and to eliminate the noise appeared while drawing, keeping only the essential characteristics. Also the hand-drawn shape may contain interruptions that must be eliminated unifying the segments. The feature extraction process is composed of a number of steps, and the first of them is the calculation of the shape's weight center.

Calculation of the shape weight center

For the calculation of the weight center of a given geometric shape are used the next formulas:

$$x_C = \frac{\sum_{i=0}^{n-1} x_i}{n}, \qquad y_C = \frac{\sum_{i=0}^{n-1} y_i}{n},$$

where x_c is the horizontal position of the shape's weight center, y_c is the vertical position of the shape's weight center, n is the number of captured points while drawing, x_i is the horizontal position of each captured point, and y_i is the vertical position of each captured point.

Extraction of significant points

The next step in the feature extraction process is the determination of the sample points. There are calculated n angularly equispaced vectors that start from the shape's weight center; n is the number of sample points. The intersection of these vectors with the boundary of the shape represents the sample points of that shape. The next step consists in the calculation and the tracing out of the tangent vectors to the shape in these points. In our application the tangent vectors are obtained by the union of the sample points; unifying two successive sample points is obtained a tangent vector.

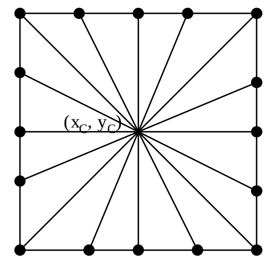

Figure 1. Extraction of sample points

As we can see in figure 1, a very important parameter in the recognizing process is the number of sample points. For an efficient extraction of the relevant information necessary for the recognition process of the geometric shape, we have to use a sufficient number of sample points.

3.2. Fuzzy Classification

To generate the input data for the neural network, after the feature extraction process follows the adaptation of the obtained information. The internal angles of a geometric shape offers the relevant information necessary to the classification process. There are calculated the angles between the consecutive tangent vectors and we obtain n angles which will be classified into four categories (fuzzy). Each angle will receive a membership value depending on the category in which it takes part, as it follows:

- 2 for the angles less than 75 degrees;
- 3 for angles between 75 and 135 degrees;
- 1 for angles between 135 and 150 degrees;
- 0 for the angles greater than 150 degrees;

The membership values must be given in the way that after the addition of the membership values according to the n angles to obtain different sums for each class of geometric shape (i. e. triangle, rectangle, circle). We have considered that the important angles are the angles less than 150 degrees. In the case of a rectangle or a triangle, along the sides we will have angles near to 180 degrees; because these angles are not significant, they receive 0 as membership value, in other words this angles will not contribute to the sum. Since the number of angles less than 150 degrees offers the relevant information necessary to the recognition process of the basic geometric shapes, only these angles, through their consistent membership values, will contribute to this sum, which will be a value from the interval [0, 3n]. Using the sum of the angles' membership values it doesn't matter the dimensions of the shape (there is no difference between a little triangle and a big one), and not even the dimensions of the sides (there is no difference between a square and a rectangle), only the internal angles matter.

3.3. Neural Recognition of the Shape

The input vector for the neural network will be obtained after the serial coding of the sum of the membership values according to the internal angles of a given geometric shape. In this way, the sum's value determines the number of bits on "1" in the serial code, and the rest of bits are "0".

The dimension of the neural network's input vector must be calculated taking into consideration the most disadvantageous case that appears when all the angles takes part of the category 3 (angles between 75 and 135 degrees). In this case the calculated sum will have the maximum value (3n), and therefore we need 3n neurons in the input layer of the neural network. Consequently the neural network's input vectors are sequences of 3n binary values. Since the neural network must recognize three categories of shapes (rectangles, triangles and circles), in the output layer we will have three neurons, one for each category. Will win the neuron with the highest output value specifying the category in which the shape takes part. Since the neural network used in this work has three layers, the dimension of the hidden layer represents a parameter and it's value will be established based on the criterion of the performances' maximization. We will vary in the next chapter the number of neurons from this layer; we want to obtain in this way the best configuration of the neural network.

4. EXPERIMENTAL RESULTS

As we specified in chapter 3.1, in the feature extraction process we have to use a number of sample points as greater as possible and this way we can extract efficiently the relevant information necessary for the recognition process of the geometric shape. But if we use too many sample points there is a risk of appearance of the noise in the extracted information. Usually the noise appears because of the undesirable hand movements while drawing with the mouse. Therefore, the number of sample points represents an other parameter that must be chosen based on the criterion of the performances' maximization. We synthesize in the table 1 the influence of this parameter on the performances of the geometric shape recognition system:

Shape [%]	**16**	**32**	**48**	**64**
Circles	63,33	100	96,66	96,66
Triangles	56,66	60	60	80
Rectangles	86,66	70	63,33	26,66
All	68,88	76,66	73,33	67,77

Table 1. The influence of the number of sample points on the shape classification.

We can see that the best number of sample points is 32.

The internal angles of the shapes are classified into four categories in the fuzzyfication stage. We showed at 3.2. that the little angles are the most important in the classification process. In chapter 3.2. we have also presented a variant of fuzzyfication. The method of according of the membership values represents the third parameter that influences the performances of the recognition system. We have studied four methods of fuzzyfication and we have to choose, depending on the results, the best solution. In the same time we have decreased the training time of the neural network from 10000 iterations to 1000. In table 2 are presented the obtained results.

Shape [%]	**I**	**II**	**III**	**IV**
Circles	100	96,66	100	100
Triangles	23,33	0	76,66	73,33
Rectangles	96,66	100	90	100
All	73,33	65,55	88,88	91,11

Table 2. Different methods of fuzzyfication of the internal angles

The four fuzzyfication methods used in this work have accorded membership values to the internal angles as follows:

Method I:
- 2 for the angles less than 75 degrees;
- 3 for angles between 75 and 135 degrees;
- 1 for angles between 135 and 150 degrees;
- 0 for the angles greater than 150 degrees.

Method II:
- 3 for angles less than 90 degrees;
- 2 for angles between 90 and 120 degrees;
- 1 for angles between 120 and 150 degrees;
- 0 for the angles greater than 150 degrees.

Method III:
- 4 for angles less then 90 degrees;
- 2 for angles between 90 and 120 degrees;
- 1 for angles between 120 and 150 degrees;
- 0 for the angles greater than 150 degrees.

Method IV:
- 5 for angles less then 90 degrees;
- 2 for angles between 90 and 120 degrees;
- 1 for angles between 120 and 150 degrees;
- 0 for the angles greater than 150 degrees.

The obtained results show that the fourth method of fuzzyfication represents the best solution.

We have continued our work by studying the influence of the neural network's architecture on the performances of the recognition system. We varied the number of neurons in the hidden layer (the first parameter) and we evaluated the recognizing rate for three different dimensions of the hidden layer: 5, 10 and 20 neurons. The obtained results are presented in table 3.

Shape [%]	**5**	**10**	**20**
Circles	96,66	100	100
Triangles	23,33	73,33	73,33
Rectangles	100	100	100
All	90	91,11	91,11

Table 3. The influence of the number of neurons from the hidden layer on the efficiency of the recognition system.

We can observe that the best solution is to use ten neurons in the hidden layer. If we increase the number of neurons in this layer over ten, the efficiency of the neural network doesn't change, but the training time grows up exponentially.

5. CONCLUSION

In this work we presented and studied a method of recognizing the basic geometric shapes. The training (recognition) is made by extracting the features from the training samples (test samples), and classifying the internal angles of the shape. The information obtained after the fuzzyfication process are used as input by a multilayer feedforward neural network. The network has learned the three classes of geometric shapes by their internal angles; the values of the internal angles are invariant in terms of scaling, translation and rotation.

We studied the influence of the number of sample points used in the feature extraction process on the efficiency of the recognition system. A small number of points (less than 32) is not sufficient for a correct recognition of the geometric shapes. In this case the information obtained in the feature extraction process are not sufficiently consistent to assure the desired performances of the recognition system. On the other hand if we use too many sample points (over 32), there is a risk of appearance of the noise in the extracted information. The noise, that usually appears because of the undesirable hand movements while drawing with the mouse, overlaps the relevant information and in this way it degrades the performances of the recognition system. The evaluations drove us to the conclusion that the best number of sample points is 32.

After that we studied different fuzzyfication methods of the internal angles. The effected tests drove us to the conclusion that the little angles (less than 90 degrees) are very important, because the number of this angles offers the relevant information necessary to the recognition system.

Finally we studied the influence of the neurons' number from the hidden layer on the efficiency of the neural network used in the recognition process. The tests shows that the best solution for the number of neurons in this layer is ten.

An unsolved problem appeared in the feature extraction process. If the obtained sample points avoid the corners of the shape, there is lost a relevant internal angle and it's replaced by other two successive angles (see figures 2 and 3).

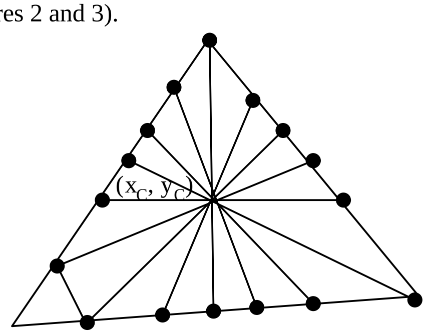

Figure 2. Extraction of sample points. One corner of the triangle is missed.

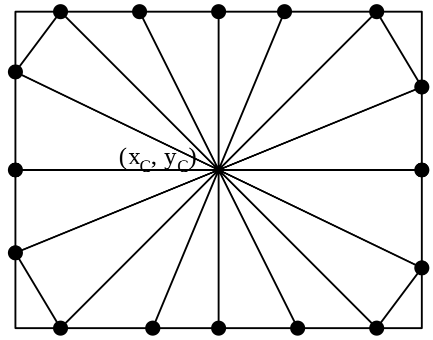

Figure 3. Extraction of sample points. All the corners of the shape are missed.

As you can see in figure 2, one corner of the triangle was replaced by other two corners and in this way, because the shape has four significant internal angles, it is possible that the recognition system will classify it as rectangle. In the same way, in figure 3. the significant angles are replaced with greater angles, and the rectangle could become circle.

In other words it is lost a part of the relevant information and it appears in addition other information (noise) which can drive to a wrong classification of the shapes. To eliminate these deficiencies, the sample points must include also the corners of the shape (if the shape has corners). This is one of the development directions of the recognition system presented in this work.

REFERENCES

Comon P., Voz J. L., Verleysen M., "Estimation of performance bounds in supervised classification", In M. Verleysen, editor, ESANN: European Symposium on Artificial Neural Networks, Bruxelles, April 1994, pag. 37 - 42.

Funahashi K. I., "On the approximate realization of continuous mappings by neural networks", Neural Networks, Vol. 2, pp. 183-192, 1989.

Guyon I., "Neural networks and applications", Internal Report AT & T Bell Labs. September 1990.

Hornik F., "Multilayer feedforward networks are universal approximators", Neural Networks, Vol. 2, pp. 359-363, 1989.

Schalkoff R.J., "Artificial Neural Networks", McGraw-Hill, 1997.

Ulgen F., Akamatsu N., Fukumi M., "On-line shape recognition with incremental training using a neural network with binary synaptic weights", Industrial Applications of NNs, CRC Press, 1999, pp. 159-192;

Zurada J.M., "Introduction to Artificial Neural Systems", West Publishing Company, St. Paul, 1992.