FUZZY NETWORKS APPLICATION FOR OCR

Josef Kokeš
Vysoká škola manažerské informatiky a ekonomiky, a.s.
Praha
Czech Republic

ABSRACT

This paper shows a concrete example of how to use a system of fuzzy logic networks for reading data from a computer screen. The system has proven particularly when the manufacturer has made deliberate steps that should make it difficult or impossible OCR from the screen. Because recognition is applied to a small set of 11 characters, the result is very fast and sufficiently reliable..

Keywords: OCR, fuzzy system, defuzzyfication.

1. FUZZY LOGIC SYSTEM

Most researchers implement their both fuzzy logic systems and neural-network applications in *Matlab*. So it is difficult to embed it into PC program, and also it would require user's knowledge about *Matlab*. In this paper we try to use Java program language to implement of Fuzzy logic using table lookup scheme for OCR. Our tool is easy to use and get a high accuracy results. This is why we used it also later, for predicting future data.

1.1 Training of Fuzzy Logic system using a table – lookup scheme

Assume, we have a set of input and output data pairs:

$$(x_1^1, x_2^1; y^1), (x_1^2, x_2^2, y^2), ... \qquad (1)$$

The task here is to generate a set of fuzzy IF-THEN rules from the given input-output pairs of (1), and then use them to determine logic system. The approach [1] divides this process into five steps as follows:

- **Step 1**: Divide each domain interval of input and output variable into 2*N+1 regions (N can be different for each variable and the width of each region can be equal or unequal), denote by SN (Small N), …, S1 (Small 1) , CE (Center), B1 (Big 1), …, BN (Big N) and then assign each region to one fuzzy membership function.

For example: We divide domain of x1, x2 and y into 5 regions (N=2) as figure 3 below.

- **Step 2**: This step further divides into three small steps. First, calculate the degree of variable in different regions. Second, assign variable to the region with maximum degree. Finally, obtain one IF-THEN rule from one pair of input-output data.

For the example above, x_1^1 has degree 0.2 in CE and 0.8 in B_1, equals to zero in all other regions. Similarly, x_1^2 has degree 1.0 in CE, equals to zero in all other regions, y^1 has 0.2 degree in CE, 0.8 degree in B_1 and equals to zero in all other regions. So x_1^1 is assigned to regions B$_1$ with 0.8 degree, x_1^2 is assigned to region CE with 1.0 degree, y^1 is assigned to region B$_1$ with 0.8 degree. Finally, we obtain rule 1:

IF (x_1 is B_1) AND (x_2 is CE) THEN y is B_1

from data pair $(x_1^1, x_1^2 \rightarrow y^1)$. Similarly, we obtain second rule 2:

IF (x_1 is S_1) AND (x_2 is B_2) THEN y is CE

from data pair $(x_1^2, x_2^2 \rightarrow y^2)$.

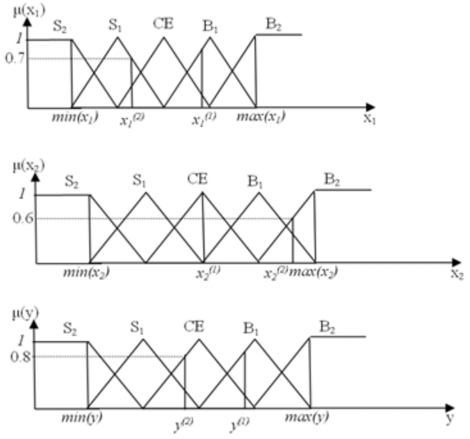


Figure 1 Demonstration of Fuzzy Logic

Table 1 Fuzzy rules

		x_1					
		S_2	S_1	CE	B_1	B_2	S_2
x_2	S_2						
	S_1						
	CE				B_1		
	B_1						
	B_2		CE				

- **Step 3**: Calculate degree of each rule based on product strategy.

$$D(rule) = \mu_A(x_1) \times \mu_B(x_2) \times \mu_C(y) \tag{2}$$

For example, to calculate degree of rule *IF (x_1 is S_1) AND (x_2 is B_2) THEN y is CE*:

$$\begin{aligned} D(rule) &= \mu_{S1}(x_1) \times \mu_{B2}(x_2) \times \mu_{CE}(y) \\ &= 0.7 \times 0.6 \times 0.8 = 0.336 \end{aligned} \tag{3}$$

where x$_1$ has degree of 0.7 in S$_1$, x$_2$ has degree of 0.6 in B$_2$ and y has degree of 0.8 in CE.

- **Step 4**: Create combine fuzzy rule base. This step avoids conflict between rules if two identical rules exist in the combine fuzzy rule base. Conflict rules are rules that have same IF part but different THEN part. To solve conflict rules, we accept only one rule from conflict group that has maximum degree. We use table – lookup presenting a fuzzy rule base. We fill cells of rule base by the rules; if there is more than one rule on one cell of fuzzy rule base, use the rule that has maximum degree.

For example, combine fuzzy rule base for rules 1 and 2 is shown in Table 1 above.

- **Step 5**: Determine mapping bases on the Combined Fuzzy Rule Base. We use the following defuzzification strategy to determine the output control y for given input (x$_1$, x$_2$). First, we combine the antecedents of the i-th fuzzy rule using product operations to determine the degree of the output control corresponding to (x$_1$, x$_2$); that is,

$$\mu_{O^i}^i = \mu_{I_1^i}(x_1) \times \mu_{I_2^i}(x_2) \tag{4}$$

where O^i denotes the output region of rule i, and I_j^i denotes the input region of rule i for the j-th component. For example, rule 2 gives:

$$\mu_{CE}^{2} = \mu_{S1}(x_1)\times\mu_{B2}(x2) \tag{5}$$

Then we use the center average defuzzification formula to determine the output:

$$y = \frac{\sum_{i=1}^{M}\mu_{O^i}^{i}\times y^{-i}}{\sum_{i=1}^{M}\mu_{O^i}^{i}} \tag{6}$$

where y^{-i} denotes the center value of region O^i and M is number of rule in the combined fuzzy rule base.

For example, for given input (x_1, x_2) and fuzzy rule base system has only two rules 1 and 2 above. Assume x_1 has 0.4 degree in region B_1, x_2 has 0.5 degree in region CE, center of region B_1 is 2.1; x_1 has 0.0 degree in region S_1 and 0.0 degree in region B_2 and center of region CE is 1.8. We can calculate out put y as following:

Firstly, we calculate output control for rule 1:

$$\mu_{B1}^{1} = \mu_{B1}(x1)\times\mu_{CE}(x2) = 0.4\times 0.5 = 0.2$$

Secondly, we calculate output control for rule 2:

$$\mu_{CE}^{2} = \mu_{S1}(x1)\times\mu_{B2}(x2) = 0.0\times 0.0 = 0.0$$

Finally, we calculate output y according to equation (6):

$$y = \frac{0.2\times 2.1 + 0.0\times 1.8}{0.2 + 0.0} = 2.1$$

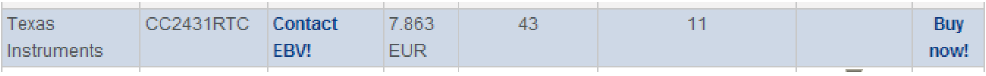
2. APPLICATION

When solving research and experimental tasks, we often encounter the problem of how to record measured data and how to keep them for later processing. The processing usually includes calibration, automatic selection of scale and basic statistical analysis and is therefore not possible to skip any phase.

Recording measured data solves an expansion unit (option) for measuring device, through which one can store all measured data to disk for later processing. Unfortunately, the financing of the project did not allow such a purchase, and therefore it was necessary to seek an alternative procedure.

An alternative solution was based on the idea that the measured values will be read from the screen using a suitable computer program for optical recognition (OCR), and subsequently stored on disk for further processing.

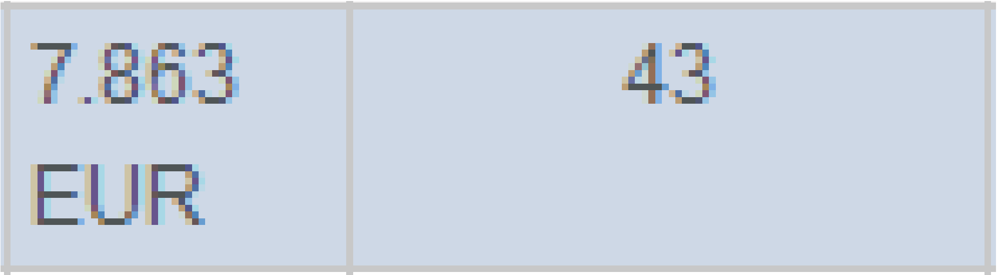
This simple procedure proved to be unfeasible. The reason is that the characters displayed on the screen are pre-distorted in various ways, so their automated reading is not easy.



Texas Instruments	CC2431RTC	Contact EBV!	7.863 EUR	43	11		Buy now!

Figure 2 Original screenshot

The same problem encountered in experiments with an automated reading data from some websites. A typical example might be trying to read the prices of commodities and their processing in order to compare offers multiple vendors. Figure 2 above shows an example (screenshot) of a display captured by such a program. Figures are perfectly readable to humans, but when deploying OCR problems arise.



7.863 EUR

43

Figure 3 Detail of characters

One can see that, for some reason, probably intentionally, data and results are visually distorted. This means that one and the same character is displayed on the screen each time differently, in a large number of variants.

Well this is shown in Figure 3, which is an enlarged detail of one section of text. It is obvious that what the human eye appears as a single character or number, in fact, is a very complicated unit composed of pixels of different colors and intensities. However, the color tone and intensity of the display changes.

In addition, probably intentionally, also character shapes are not always exactly the same. This is why we cannot use simple threshold, and then converting to black and white scale. Attempts to solve these problems led to the idea, to put on for the OCR neural network and train it to recognize numbers 0-9 and the character "dot".

3. IMPLEMENTATION

To implement this algorithm, we use Hash Map (HM) in Collection Frame Work of Java to implement fuzzy rule base. It is illustrated by figure 4 below.

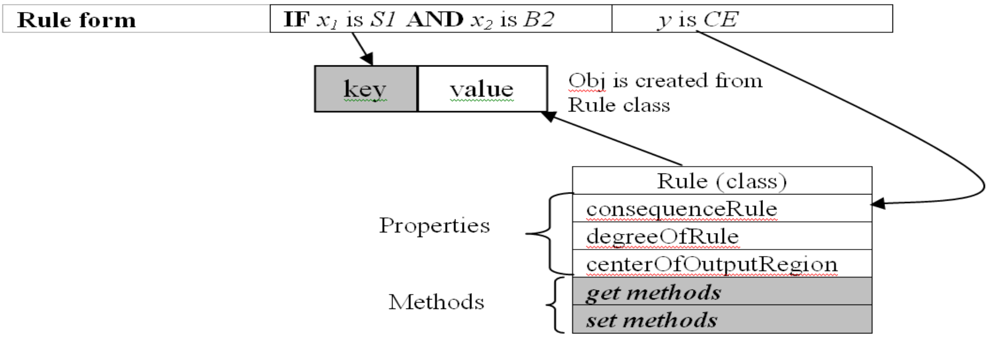


Figure 4 Schema of algorithm used

Each element of HM consists of two parts. The first part is the key and second part is the value correspon-ding to this key. We choose the key is the antecedents of the rule to avoid conflicting rules. For easy defuzzification and compare degree of two rules have the same IF part we choose value in object that is created from class Rule.

4. CONCLUSION

Described application of fuzzy sets resolved all problems with the OCR reading texts on the PC screen. The system is fully proven. Because of its use for measuring purposes is no longer needed, it was re-trained again. Nowadays, I use it to read and record the values of www.forex.com.

5. REFERENCES

[1] WANG, Y.F., Predicting stock price using fuzzy grey prediction system, Expert system with application 22 (2002) 33-39.

[2] COSTA, B.S.J., BEZERRA C.G., OLIVEIRA, L.A.: Java Fuzzy Logic Toolbox for Industrial Process Control. http://www.dca.ufrn.br/~affonso/FTP/artigos/2010/cba2010_fuzzy.pdf

[3] ROSS, T. J. (2010) Front Matter, in Fuzzy Logic with Engineering Applications, Third Edition, John Wiley & Sons, Ltd, Chichester, UK. doi: 10.1002/9781119994374.fmatter

***Acknowledgement**: This work is supported by grant FR-TI2_643_v2.*