

About the NeuroFuzzy Module of the FuzzyTECH5.5 Software

Ágnes B. Simon, Dániel Biró

College of Nyíregyháza, Sóstói út 31, simona@nyf.hu, bibby@freemail.hu

Abstract: Our online edition of the software FuzzyTECH5.5 has a NeuroFuzzy Module. We studied it through the simulation sample program of the recycling glass classification. We collected the experiences to develop a fuzzy logic system based on training data: We need to have the sample data in type of .csv file, we have to develop an “empty” fuzzy logic system, we must choose the CoM compromising defuzzification method from the offers. The NeuroFuzzy modul gives us opportunity to open both rules and membership functions for learning. After the last manual optimization we have a perfect fuzzy logic system.

1 Introduction

To expand the capabilities of *fuzzyTECH*, the INFORM GmbH added several modules to the online edition. The *NeuroFuzzy Module* uses neural network technology to generate fuzzy logic rules and membership functions automatically.

Our online edition of this software (FuzzyTECH5.54j) has this module. We studied it through the simulation sample program of the recycling glass classification.

In the next sections we describe our experiences.

2 Integration the Benefits of the Fuzzy Logic and the Neural Networks

Using the project editor of the program *fuzzyTECH* we can design a fuzzy logic system which has all the knowledge about the system including the human experience and the fuzziness we are not able to describe by exact differential equations.

We can specify a fuzzy logic controlling system by a lot of rules having the form IF <situation> THEN <action>. Both the <situation> and <action> are built up from linguistic variables combined with the appropriate operators.

To develop a fuzzy logic control system we need to take three computational steps: fuzzification, fuzzy inference and defuzzification. We are lucky if the all knowledge about a system we want to automate we can build into the system project directly.

There are applications where the know-how is not explicit so we need to learn it from the data set coming from the behavior of the system.

The neural networks are able to learn from data to set up a system.

In neural nets, the parameters affected by the training procedure are the connection weights between the neurons. Looking at a fuzzy system using arbitrary sets of fuzzy rules, the firing strength of each rule output connects this rule with the precondition of the following rule block. Using the technology of Fuzzy Associative Maps (FAM), this connection can be weighted. With FAM, each rule is assigned a Degree of Support(DoS) representing the individual importance of the rule. Rules themselves can be “fuzzy” – meaning, with a support between 0 and 1. This rule plausibility can be used as the parameter to be calibrated in the neural training. A neural training algorithm can calculate an error gradient by slightly changing the rule weights.

Integrating the explicit knowledge representation of the fuzzy logic with the learning power of the neural networks we can get the NeuroFuzzy technology.

The NeuroFuzzy module provides methods for supervised learning. The heuristic methods used combine the two learning steps of error backpropagation with the idea of competitive learning. After a system output is computed by forward propagation, an output error is identified by comparing the system output with the given sample output data.

The standard methods perform an optimization step per sample. Batch methods calculate errors and gradients for all samples and update the system's parameters after each complete iteration. Standard methods provide a better computation performance and batch methods a better convergence behavior. *FuzzyTECH* also allows us to include user defined method by writing the training method as a dynamic link library.

The NeuroFuzzy modul gives us opportunity to open both rules and membership functions for learning.

Before training is started, the membership functions of all open terms of a linguistic variable are converted standard types: Inputs are converted to Z, Lambda and S forms. For the output variables, all terms are converted to Lambda type membership functions.

In addition, the standard variables are restricted to shapes which allow an overlap of no more than two terms. This leads to a variable definition in which each term is determined by exactly one parameter, the base value of the term's maximum. As a result, each term must intersect the base line at the maximum point of its neighbours (On view Figure 131 in [1]).

3 Using the NeuroFuzzy Module

Among the samples program of our FuzzyTECH edition there is a simulation program for separating the color bottles in a glass recycling plant.

This program can simulate the work of the human operator selecting the bottles into different containers recording the color intensity of the bottles and the type numbers of the containers where he put the bottles. Using his know-how we can get a training data to develop a project which can automate this process by using the NeuroFuzzy Modul.

In addition, the simulation includes the option for testing the generated fuzzy logic system.

We used this simulation to learn the steps of NeuroFuzzy System Design using this modul in the fuzzyTECH5.5 online edition.

3.1 Neuro Fuzzy System Design

3.1.1 Creating the Training Data

Before NeuroFuzzy training we need to have the sample data in type of .csv file. In this case we obtained this type of training data file by initiating the simulation program in Record mode. Figure 1 shows a part of the file Nfsensor.csv.

Record_#	BlueRed	GreenBlue	RedGreen	Type
Record_#1,	1739.94,	1675.15,	2395.96,	3
Record_#2,	373.30,	3087.05,	2429.47,	4
Record_#3,	1756.77,	1652.00,	1514.98,	3
Record_#4,	864.45,	1647.31,	2665.90,	1
Record_#5,	222.85,	3059.54,	2002.33,	4
Record_#6,	877.88,	2031.66,	3071.18,	1
Record_#7,	1803.58,	1583.12,	2163.05,	3
Record_#8,	2352.12,	2557.04,	1411.53,	2
Record_#9,	401.30,	3259.94,	2150.98,	4
Record_#10,	363.34,	3477.95,	2462.86,	4
Record_#11,	1571.17,	1731.04,	1735.33,	3
Record_#12,	104.80,	3389.83,	2421.83,	4
Record_#13,	499.85,	3305.75,	2196.22,	4
Record_#14,	2297.28,	3340.14,	535.62,	2
Record_#15,	2092.62,	3177.21,	584.32,	2

Figure 1
Viewing the recorded samples in the WordPad editor

The color intensities of the bottle (BlueRed, GreenBlue, RedGreen) and the type of the container in which the human operator put it (Type1=red, Type2=green, Type3=white, Type4=blue) to be melted for making new glass, are set down in every record of the file.

If we had obtained the sample data we had to develop an “empty” initial fuzzy logic system for learning.

3.1.2 Creating an Initial Fuzzy Logic System Based on the Sample Data

Setting up a fuzzy logic system may be by Fuzzy Design Wizard (FDW) quickly.

We had to sign the "Use a data file" checkbox in the first dialog window. (Fig. 2)

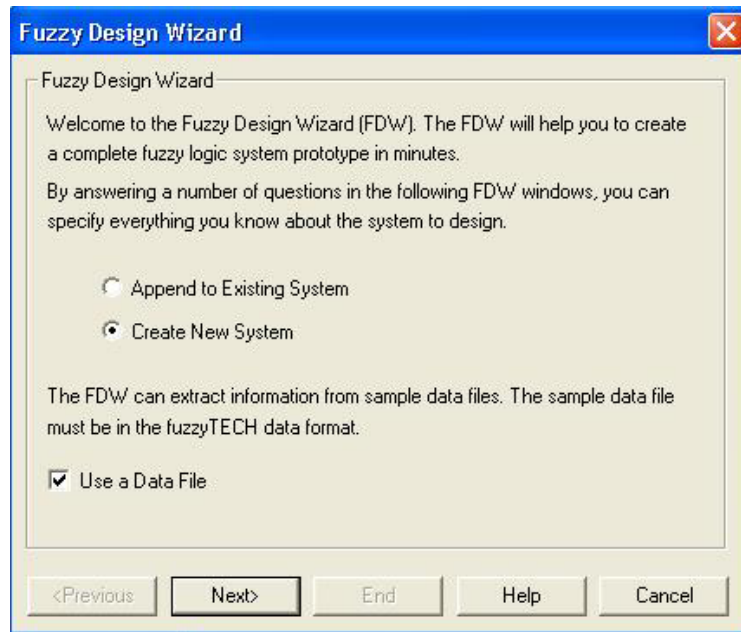


Figure 2
Fast designing a fuzzy logic system based on the sample data file

After opening the appropriate sample data file, we can see and modify the screens of „Define variable”, „Define defuzzification”, „Define rule blocks”.

The FDW extracts from the file:

- a) . the number of variables,
- b) . the names of the variables,
- c) . the data range of each variable.

The FDW selects defaults for the fuzzy prototype system with assumptions below:

- 1) the variables being to the last variable are inputs;
- 2) the last variable will be the output,
- 3) . all inputs can be represented by three terms,
- 4) . the output can be represented by five terms (in our case we modified it to four terms)
- 5) default names are used for the terms, (terms of output variable were renamed in hungarian: piros, zöld, feher, kék)

- 6) CoM defuzzification is used for the output,
- 7) all variables are inferred via a single rule block, and
- 8) a complete rule base is created by using a DoS =0 value for all rules.

When we met the question *Do you want to generate the specified project?*, we answered „Yes” for it, and we obtained an initial fuzzy logic system (Figure 3).

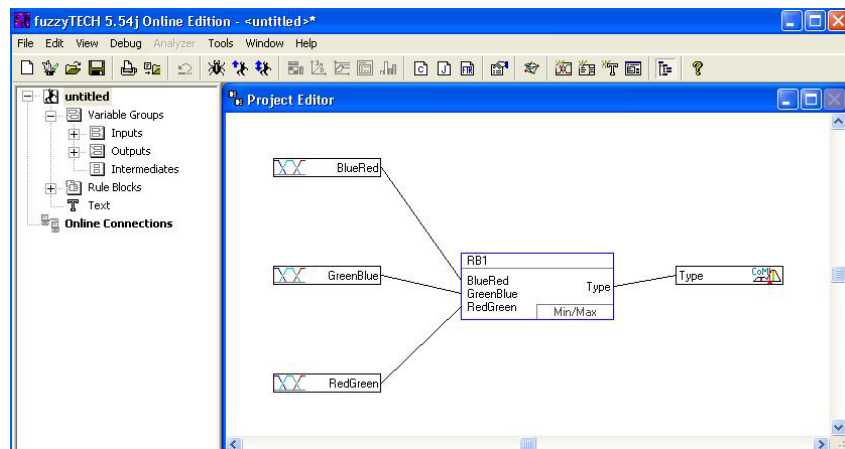


Figure 3

The structure of generated fuzzy logic system based on Nfsensor.csv

Let see the rules by double-clicking on the graphic element RB1 in the "Project Editor" window! (Figure 4). The rules all have the weight of zero, that means there is not yet knowledge in the system. The knowledge, the rules for the glass sorting process, gets into the system by the NeuroFuzzy training step.

In the column of DoS the dark backgrounds of the values display they will be learning.

We had to choose the CoM compromising defuzzification method because the NeuroFuzzy training uses error gradients to determine a direction of ptimization.

The response of the system to small parameter changes is evaluated to determine the gradients. The non compromising defuzzification methods do not react to small changes (in [1] p. 96).

#	IF	THEN	DoS	Type	
1	low	low	low	0.00	piros
2	low	low	low	0.00	zold
3	low	low	low	0.00	feher
4	low	low	low	0.00	kek
5	low	low	medium	0.00	piros
6	low	low	medium	0.00	zold
7	low	low	medium	0.00	feher
8	low	low	medium	0.00	kek
9	low	low	high	0.00	piros
10	low	low	high	0.00	zold
11	low	low	high	0.00	feher
12	low	low	high	0.00	kek
13	low	medium	low	0.00	piros
14	low	medium	low	0.00	zold
15	low	medium	low	0.00	feher
16	low	medium	low	0.00	kek
17	low	medium	medium	0.00	piros
18	low	medium	medium	0.00	zold

Figure 4
Rule block before learning

3.1.3 Entering A-priori Knowledge

The project editor allow us to make changes in the default parameters of the project enters. So we modified the names of the terms “Type” linguistic variables from English into Hungarian.

3.1.4 Opening the Components for Learning

For the linguistic variables we can open specific terms for training, and for the rule blocks we can open the individual rules for training by buttons marked with *L* letter in the small icons.

3.1.5 Starting the Training

To initiate NeuroFuzzy training, we selected Tools/Neuro/Learning from the main menu of fuzzyTECH. Selecting the file *Nfsensor.csv* in the "Read Example File..." dialog box, the "Learn Control" dialog box opened. The tool bar of this dialog box lets us control the NeuroFuzzy processing.

In the Figure 5 we can see the report of a learning phase.

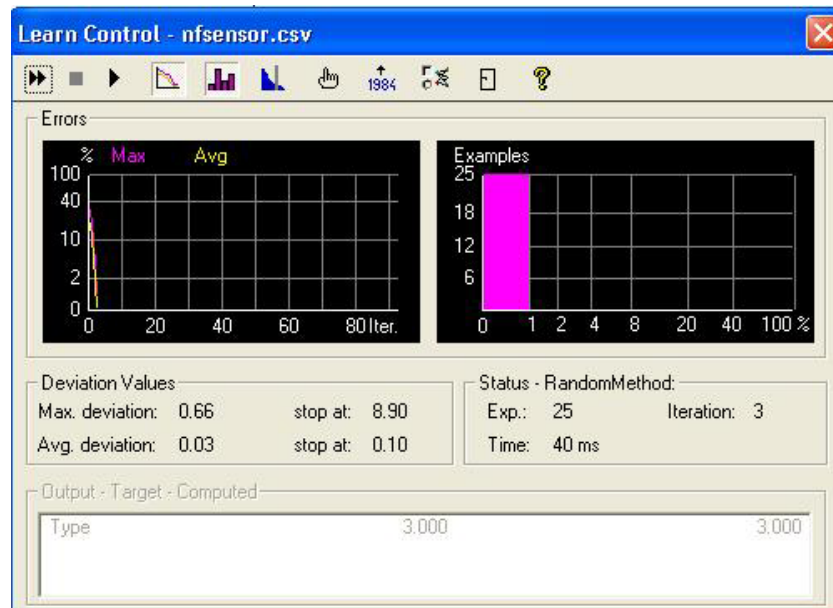


Figure 5
Learn Control window

The error plot shows the deviation of the worst sample (Max.) and the average deviation of all samples (Avg.) for every iteration.

The statistic plot classifies the deviations of all samples. In each class, the number of samples matching the deviation class is counted. In our case Max. deviation and Avg. deviation started at 30% and lowered to 1%. The percentage is calculated based on the base variable range of the output variable. The training was terminated because the average deviation less then 0.05. The range of the "Type" is 5.

The status section displays the number of the currently treated sample, the number of completed iterations, and the computation time for the last computation. The field indicates the time for a complete iteration during continuous training and the time for the last training step while in step mode.

The deviation section shows the current values for the deviation of the worst sample and the average deviation. The section displays different values when the NeuroFuzzy module is used in continuous training or when single training steps are taken.

3.1.6 Final Optimization

The result of NeuroFuzzy training is a fuzzy logic system that we can directly optimize by hand. How to manually optimize the system depends on the application. Often the objective of the training is to find functionally redundant rules or unnecessary rules.

Conclusions

The FuzzyTECH5.5 program online edition is very useful program to develop an adaptiv fuzzy logic system by using the NeuroFuzzy Modul.

The most important steps to set up the training and open the learning:

- A complete fuzzy logic system must be designed.
- Components of the system for training must be determined.
- A training configuration must be construct.
- A sample file must be loaded.

The excellent samples and the manual help us to go forward in practice and we can test the results by the simulation program delivered also.

References

- [1] User's Manual for all *fuzzyTECH* 5.5 Editions
- [2] *András Botos, Ágnes B. Simon*, First Steps in using of FuzzyTech5.5, Online Edition at College of Nyíregyháza, Proceedings of 3rd International Symposium of Hungarian Reseachers on Computational Intelligence, Budapest, 2002
- [3] *Simonné dr. Balogh Ágnes*, Lány számítástechnikai módszerek alkalmazásának lehetősége a FuzzyTECH5.5 szoftverrel. Agrárinformatikai Nyári Egyetem és Fórum, Gödöllő, 2004
- [4] *Simonné dr. Balogh Ágnes, Biró Dániel*, A neurális hálók és a fuzzy logika kombinálása, Főiskolák matematika, fizika és számítástechnika tanárainak XXVIII. konferenciája, Nyíregyháza, 2004