

# Learning Methods for Fuzzy Systems

Rudolf Kruse and Andreas Nürnberger

*Department of Computer Science, University of Magdeburg*

*Universitätsplatz 2, D-39106 Magdeburg, Germany*

*Phone : +49.391.67.18706, Fax : +49.391.67.12018*

*E-Mail: kruse@iik.cs.uni-magdeburg.de*

**Abstract.** Fuzzy systems are currently being used in a wide field of industrial and scientific applications. Therefore, it is necessary to have algorithms which construct and optimize such systems automatically. Since the idea of learning is being studied in other research areas like machine learning and data mining, some of the developed methods have been made available and optimized for the learning process in fuzzy systems. In this paper, we present a short survey of these methods and take a closer look at a special learning approach, the neuro-fuzzy systems.

## 1. Introduction

At present, fuzzy systems are being used in a wide range of industrial and scientific applications with the main application areas being fuzzy control, data analysis and knowledge based systems. Fuzzy controllers, for instance, model the control strategy of a human expert to control a system for which no mathematical or physical model exists. They employ a set of linguistic rules to describe the human behavior. A rule in the domain of speed control, for example, could have the form:

*'If the speed is very high and the distance is small, then brake very strongly'.*

The linguistic rules describe a control surface, which defines an appropriate output value for every vector of input values. Thus, a function is defined which 'fits' the rules.

The major benefits of fuzzy techniques are the convenient method to model technical systems and the good interpretability of the system description by using linguistic rules. However, the implementation of a fuzzy system can be very time consuming because there are no formal methods to determine its parameters (fuzzy sets and fuzzy rules). Therefore it is necessary to have algorithms which can learn fuzzy systems automatically from data.

Since the idea of learning is being studied in other research areas like machine learning and data mining, it is obvious to make the developed methods available for the learning process in fuzzy systems. The most important methods are currently derived from statistics [8], cluster analysis [4] or neural network theory [15]. All methods use sample data (data vectors, observation data) to learn from.

In this paper we present a short introduction to learning methods for fuzzy systems and discuss one approach in more detail: the neuro-fuzzy systems.

## 2. Fuzzy Systems

Different methods have been developed to automatically construct fuzzy systems from data. Most of them are based on the principle to construct a fuzzy system for function ap-

proximation. Such a system consists of  $r$  parallel rules. Let  $x_1, \dots, x_n$  be  $n$  input values,  $y$  the output value and  $\mu_r^{(1)}, \dots, \mu_r^{(n)}$  and  $v_r$  fuzzy sets, then the fuzzy rules of a fuzzy system can be defined as:

$$\text{if } x_1 \text{ is } \mu_r^{(1)} \text{ and } x_2 \text{ is } \mu_r^{(2)} \text{ and } \dots \text{ and } x_n \text{ is } \mu_r^{(n)} \text{ then } y \text{ is } v_r .$$

To calculate the output of this system, the output of every rule is computed first. Then, all outputs are combined into a single system output (usually, a fuzzy set). A *crisp* (exact) output value is derived by a defuzzification procedure [8], e.g. the center of gravity of the resulting fuzzy set. A fuzzy system defined in this way approximates an unknown function based on vague samples, which are described by the fuzzy sets. To construct or learn a fuzzy system, the fuzzy rules as well as the membership functions describing the fuzzy sets have to be defined. This can be done by using learning methods like cluster analysis, neuro-fuzzy approaches or a combination of them.

Cluster analysis is a technique for classifying data, i.e. to divide given data into sets of classes or *clusters*. In ‘classical’ cluster analysis each datum has to be assigned to exactly one class. This strict division is often not applicable to real world problems. Fuzzy cluster analysis relaxes this requirement by allowing gradual memberships. This offers the opportunity to deal with data that belong to more than one class. A survey to fuzzy clustering algorithms is presented in [4].

By using fuzzy clustering methods it is possible to learn fuzzy *if-then* rules from data. An example of learned fuzzy rules, which assigns one output value to two input values, is presented in Figure 1. The data points (depicted in the left top of Figure 1) are connected to the cluster for which they achieve the highest membership value. Every cluster represents a

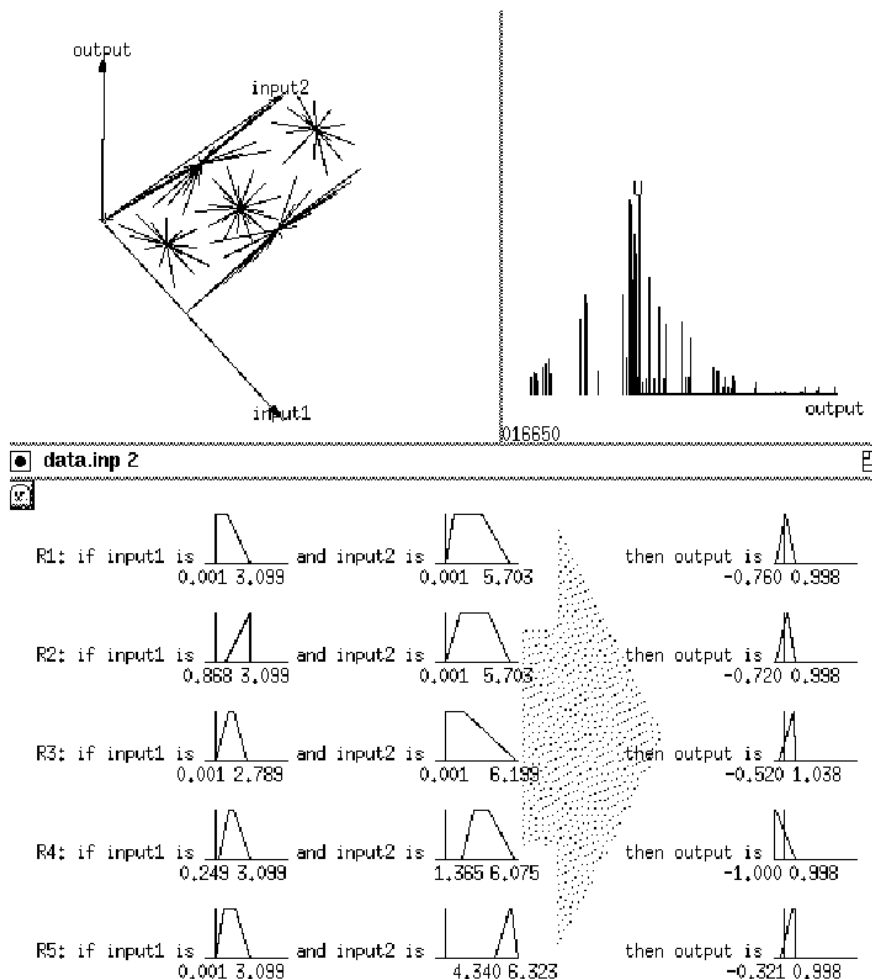


Figure 1. An example of fuzzy cluster analysis created with the tool FCLUSTER

fuzzy *if-then* rule and can be seen as a multidimensional fuzzy set. A fuzzy set in a single dimension is derived by projecting a cluster to one dimension (see the graph in the right top of Figure 1). The presented 5 rules can be obtained by projecting all clusters and smoothing the projection to triangular or trapezoidal fuzzy sets. Using these techniques, for example, fuzzy controllers and fuzzy approximation systems can be constructed [5].

A major drawback of this approach is that a rulebase derived by fuzzy clustering methods is often not easy to interpret, since no restrictions concerning the form of the fuzzy sets can be defined. Further, prior knowledge given by already existing fuzzy rules is not easy to introduce into the learning process. Neuro-fuzzy systems present a solution to these problems. They can be used to learn fuzzy rules and fuzzy sets, but also to optimize fuzzy systems derived by fuzzy clustering algorithms.

### 3. Neuro-Fuzzy Systems

Neuro-fuzzy systems are fuzzy systems that are trained by a learning algorithm derived from neural network theory. The (heuristic) learning procedure operates on local information, and causes only local changes to the underlying fuzzy system. The learning process is not knowledge based, but data driven.

A neuro-fuzzy system can be viewed as a special 3-layer feedforward neural network. The units in this network use t-norms or t-conorms instead of the activation functions common in neural networks. The first layer represents input variables, the middle (hidden) layer represents fuzzy rules, and the third layer represents output variables. Fuzzy sets are encoded as (fuzzy) connection weights. Some neuro-fuzzy models use more than 3 layers, and encode fuzzy sets as activation functions. It is usually possible to transform these models into a 3-layer architecture. This view of a fuzzy system illustrates the data flow within the system and its parallel nature. However, this neural network view is not a prerequisite for applying a learning procedure, it is merely a convenience.

A neuro-fuzzy system can always (i.e. before, during and after learning) be interpreted as a system of fuzzy rules. It is both possible to create the system out of training data from scratch, and it is possible to initialize it by prior knowledge in form of fuzzy rules.

The learning procedure of a neuro-fuzzy system takes the semantic properties of the underlying fuzzy system into account. This results in constraints on the possible modifications applicable to the system's parameters.

Neuro-fuzzy systems approximate n-dimensional (unknown) functions that are partially given by the training data. The fuzzy rules encoded within the system represent vague samples, and can be viewed as vague prototypes of the training data. A neuro-fuzzy system should not be seen as a kind of (fuzzy) expert system, and it has nothing to do with fuzzy logic in the narrow sense [8]. An approach to fuzzy expert systems is realized by possibilistic networks [3].

Therefore, neuro-fuzzy can be considered as a specific technique to derive a fuzzy system from data, or to enhance it by learning from examples. The exact implementation of the neuro-fuzzy model is of no importance. It is possible to use a neural network to learn certain parameters of a fuzzy system, like using a self-organizing feature map to find fuzzy rules [18] (cooperative models), or to view a fuzzy system as a special neural network, and directly apply a learning algorithm [12] (hybrid models).

An example of a hybrid neuro-fuzzy system is the NEFCON model. It has been originally developed to learn and optimize the rulebase of a fuzzy controller [11], but it can also be seen as a neuro-fuzzy system for function approximation.

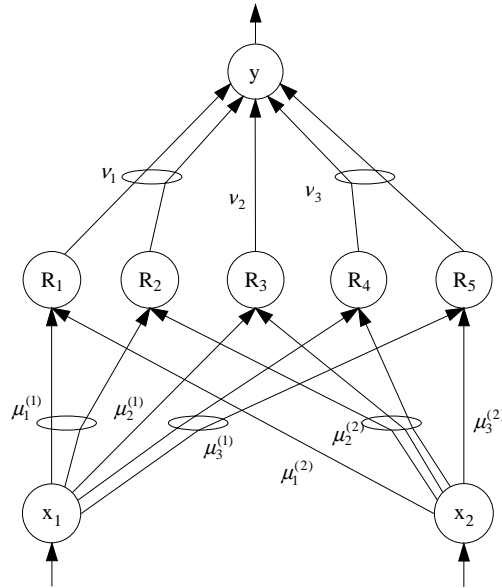


Figure 2. A NEFCON system with two inputs, 5 rules and one output

#### 4. The NEFCON-Model

The NEFCON-Model is based on a three layer generic fuzzy perceptron [13; 15]. Fig. 2 presents an example of a NEFCON system, which describes the structure of a fuzzy system with 5 rules, 2 inputs, and one output. The inner nodes  $R_1, \dots, R_5$  represent the rules, the nodes  $x_1, x_2$ , and  $y$  the input and output values, and  $\mu_r^{(i)}, v_r$  the fuzzy sets describing the antecedents  $A_r^{(i)}$  and conclusions  $B_r$ . Rules with the same antecedents use so-called shared weights, which are represented by ellipses in Fig. 2. They ensure the integrity of the rulebase. Node  $R_1$ , for example, represents the rule:  $R_1$ : if  $x_1$  is  $A_1^{(1)}$  and  $x_2$  is  $A_1^{(2)}$  then  $y$  is  $B_1$ .

The learning process of the NEFCON model can be divided into two main phases. The first phase aims at finding an appropriate initial rulebase as soon as possible. If no prior knowledge is available, the rulebase is learned from scratch. If, on the other hand, a manually defined rulebase is already given, the algorithm completes this rulebase. In the second phase, the acquired rulebase is optimized by shifting or modifying the fuzzy sets of the rules. Both phases use a fuzzy error, which describes the quality of the current system state, to learn or to optimize the rulebase. The fuzzy error can be derived by calculating the difference to the desired output (in case of function approximation, if the output is known), or by a linguistic error definition (in case of fuzzy control). In case of a linguistic error description, the linguistic rules are used to describe ‘good’ and ‘bad’ situations of the dynamic system, which has to be controlled [15]. So the system is able to learn online, since no input/output pairs must be given. The fuzzy error plays the role of a critic element in reinforcement learning models (e.g. [1; 2]).

#### 5. An Implementation of the NEFCON Model

The NEFCON model has been implemented in different environments. A recent implementation was done under MATLAB/SIMULINK<sup>1</sup> with updated learning methods for the

<sup>1</sup> MATLAB/SIMULINK is a simulation tool developed and distributed by ‘The Mathworks’ Inc., 24 Prime Park Way, Natick, Mass.01760; WWW: <http://www.mathworks.com>.

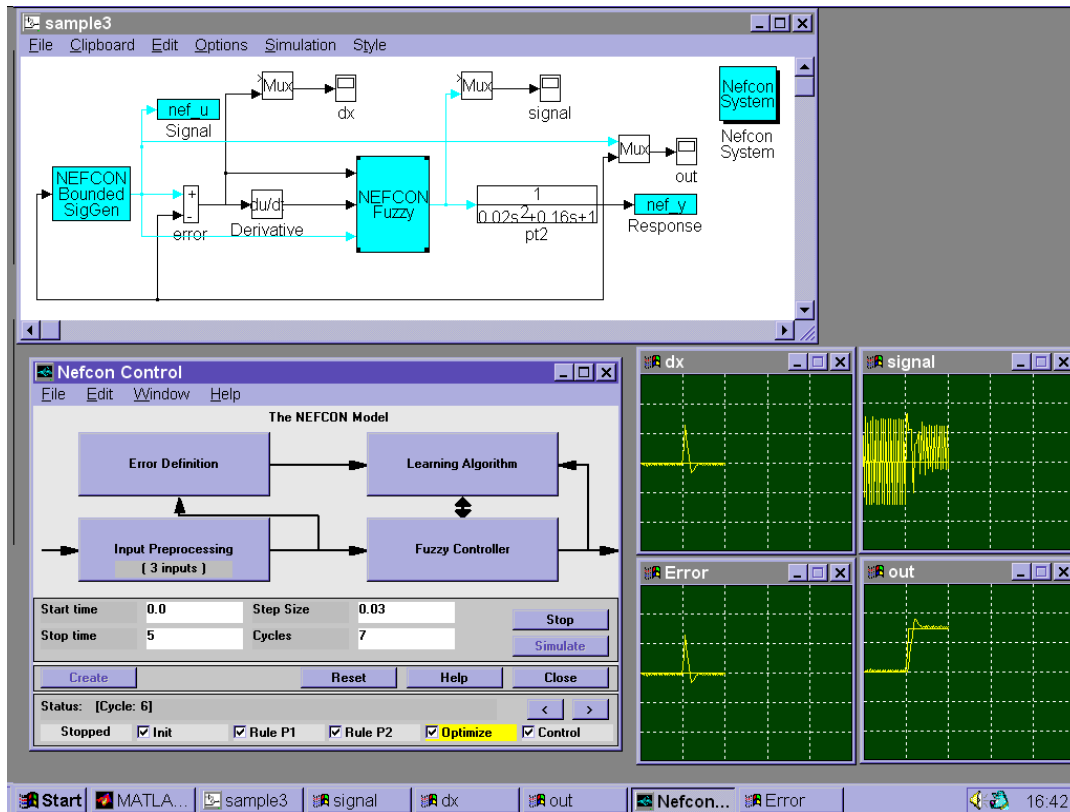


Figure 3. Sample of a development environment under MATLAB/SIMULINK (PT<sub>2</sub>-System)

development of fuzzy controllers in industrial research applications. The NEFCON learning algorithms [14; 16] learn and optimize the rulebase of a Mamdani-like fuzzy controller [10]. The fuzzy sets of the antecedents and conclusions can be represented by any symmetrical membership function.

The major goal of the implementation of the NEFCON model under MATLAB/SIMULINK was to provide an interactive tool for the construction and optimization of a fuzzy controller. This tool enables the user to include prior knowledge into the fuzzy system, to stop and to resume the learning process at any time, to modify the rulebase and the optimization parameters interactively, and to define the fuzzy error in a convenient way. Figure 3 presents the simulation environment of a sample application. It was created under Microsoft Windows NT 4.0.

The simulation results concerning a conventional PT<sub>2</sub> system (see [9; 6]) during the learning cycles are presented in Figure 4. The learning algorithm started with three equally distributed trapezoidal fuzzy sets for each input variable and five for the output variable. The fuzzy error was described using 'fuzzy intervals' [16]. The corresponding simulation

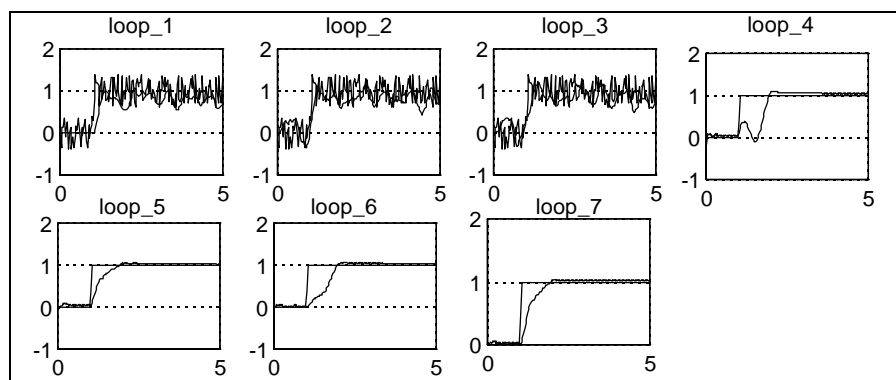


Figure 4. Simulation results for a PT<sub>2</sub> system

environment is shown in Figure 3. The algorithm used a noisy reference signal during rule learning to improve the coverage of the system state space (see cycles 1-3 in Figure 4). The system was able to find an appropriate rulebase within 3 rule learning and 3 optimization cycles (with 167 iteration steps each cycle). The optimized rulebase consists of 25 rules.

## 6. Conclusion

In this paper we presented some methods of learning fuzzy systems from data. However, the interpretability of learned or optimized fuzzy systems can be sometimes poor when there are no constraints applied to the learning process. Learning methods should be seen as an addition to manually defined and (semantically) clearly structured fuzzy systems. Such systems can be more easily maintained and checked for plausibility.

The tools mentioned in this paper, in particular FCLUSTER and NEFCON, as well as further information concerning fuzzy systems, are available on the Internet at: <http://fuzzy.cs.uni-magdeburg.de>.

## References

- [1] Barto, A.G., Sutton R. S., Anderson, C. W. (1983): Neuronlike adaptive elements that can solve difficult learning control problems, *IEEE Transactions on Systems, Man and Cybernetics*, 13:834-846
- [2] Barto, A.G. (1992): Reinforcement Learning and Adaptive Critic Methods, In [17]
- [3] Gebhardt, Jörg and Kruse, Rudolf (1996): Learning Possibilistic Networks from Data, in D. Fisher, H. Lenz (eds.): *Learning from Data, Artificial Intelligence and Statistics 5, Lecture Notes in Statistics 112*, 143-153, Springer, New York
- [4] Höppner, Frank; Klawonn, Frank; Kruse, Rudolf (1996): *Fuzzy-Clusteranalyse, Computational Intelligence*, Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig, Wiesbaden
- [5] Klawonn, Frank and Kruse, Rudolf (1997): Constructing a Fuzzy Controller from Data, In *Fuzzy Sets and Systems* 85
- [6] Knappe, Heiko (1994): Comparison of Conventional and Fuzzy-Control of Non-Linear Systems, in [7]
- [7] Kruse, Rudolf; Gebhardt, Jörg; Palm, Rainer (Eds.) (1994): *Fuzzy Systems in Computer Science*, Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig, Wiesbaden
- [8] Kruse, Rudolf; Gebhardt, Jörg, Klawonn, Frank (1994): *Foundations of Fuzzy Systems*, John Wiley & Sons, Inc., New York, Chichester, et.al.
- [9] Leonhard, Werner (1992): *Einführung in die Regelungstechnik*, Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig, Wiesbaden
- [10] Mamdani, E. H.; Assilian S. (1973): An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller, *International Journal of Man-Machine Studies*, 7:1-13
- [11] Nauck, Detlef and Kruse, Rudolf (1993): A Fuzzy Neural Network Learning Fuzzy Control Rules and Membership Functions by Fuzzy Error Backpropagation, In *Proc. IEEE Int. Conf. on Neural Networks 1993*, San Francisco
- [12] Nauck, Detlef and Kruse, Rudolf (1996): Designing neuro-fuzzy systems through backpropagation, In Witold Pedryz, editor, *Fuzzy Modelling: Paradigms and Practice*, pages 203-228, Kluwer, Boston
- [13] Nauck, Detlef (1994): A Fuzzy Perceptron as a Generic Model for Neuro-Fuzzy Approaches, In *Proc. of the 2nd German GI-Workshop Fuzzy-Systeme '94*, München
- [14] Nauck, Detlef; Kruse, Rudolf; Stellmach, Roland (1995): New Learning Algorithms for the Neuro-Fuzzy Environment NEFCON-I, In *Proceedings of Neuro-Fuzzy-Systeme '95*, 357-364, Darmstadt
- [15] Nauck, Detlef; Klawonn, Frank; Kruse, Rudolf (1997): *Foundations of Neuro-Fuzzy Systems*, John Wiley & Sons, Inc., New York, Chichester, et.al. (to appear)
- [16] Nürnberger, Andreas; Nauck, Detlef; Kruse, Rudolf; Merz, Ludger (1997): A Neuro-Fuzzy Development Tool for Fuzzy Controllers under MATLAB/SIMULINK, to appear in: *Proc. of the 5th European Congress on Intelligent Techniques and Soft Computing (EUFIT'97)*, Aachen
- [17] White, D. A., Sofge, D. A., Hrsg. (1992): *Handbook of Intelligent Control. Neural, Fuzzy and Adaptive Approaches*, Van Nostrand Reinhold, New York
- [18] Witold Pedryz and Card, H.C (1992): Linguistic interpretation of self-organizing maps, In *Proc. IEEE Int. Conf. On Fuzzy Systems 1992*, pages 371-378, San Diego