What are Neuro–Fuzzy Classifiers?

Detlef Nauck and Rudolf Kruse
University of Magdeburg
Faculty of Computer Science
Institute of Information and Communication Systems
Universitaetsplatz 2, D–39106 Magdeburg, Germany
Phone: +49.391.67.12700, Fax: +49.391.67.12018
E-Mail: nauck@iik.cs.uni-magdeburg.de, URL: fuzzy.cs.uni-magdeburg.de/∼nauck

Keywords: fuzzy classification, learning, neuro–fuzzy system

Abstract

Neuro–fuzzy combination are considered for several years already. However, the term "neuro–fuzzy" still lacks of proper definition, and it has the flavor of a "buzz word". In this paper we try to give it a meaning in the context of fuzzy classification systems. From our point of view "neuro–fuzzy" means the employment of heuristic learning strategies derived from the domain of neural network theory to support the development of a fuzzy system. We illustrate our ideas using our NEFCLASS model which is used to create a fuzzy classification system from data.

1 Introduction

Ever since fuzzy systems were applied in industrial applications, developers know that the construction of a well performing fuzzy system is not always easy. The problem of finding appropriate membership functions and fuzzy rules is often a tiring process of trial and error. Therefore the idea of applying learning algorithms to fuzzy systems was considered early Approaches to so called *adaptive* or *self–organizing fuzzy controllers* can be found e.g. in [18, 19, 20]. An overview of this area is presented in [5]. These kind of adaptive models usually use knowledge based methods. However, another possibility of learning parameters of fuzzy systems is given by neural networks.

The learning capabilities of neural networks made them a prime target for a combination with fuzzy systems in order to automate or support the process of devloping a fuzzy system for a given task. The first so–called neuro–fuzzy approaches were considered mainly in the domain of (neuro–) fuzzy control, but today the approach is more general. Neuro–fuzzy systems are applied in various domains, e.g. control, data analysis, decision support, etc.

Modern neuro–fuzzy–systems are usually represented as a multilayer feedforward neural network [1, 3, 4, 7, 10, 12, 13, 21], but fuzzifications of other neural network architectures are also considered, for example self–organizing feature maps [2, 22]. In neuro–fuzzy models, connection weights and propagation and activation functions differ from common neural networks. Although there are a lot of different approaches [3, 13], we want to restrict the term "neuro–fuzzy" to systems which display the following properties:

(i) A neuro–fuzzy system is a fuzzy system that is trained by a learning algorithm (usually) derived from neural network theory. The (heuristical) learning procedure operates on local information, and causes only local modifications in the underlying fuzzy system. The learning process is not knowledge based, but data driven.

(ii) A neuro–fuzzy system can be viewed as a spe-

This paper appeared in Proc. Seventh International Fuzzy Systems Association World Congress IFSA'97, Vol. IV, pp. 228–233, Academia Prague, 1997.

cial 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. 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.

(iii) 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.

(iv) The learning procedure of a neuro–fuzzy system takes the semantical properties of the underlying fuzzy system into account. This results in constraints on the possible modifications applicable to the system parameters.

(v) A neuro–fuzzy system approximates an n–dimensional (unknown) function that is 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 [9].

In this paper "neuro–fuzzy" has to be understood in the way given by the five points above. Therefore we consider "neuro–fuzzy" as a certain technique to derive a fuzzy system from data, or to enhance it by learning from examples. The exact implementation or "neuro–fuzzy model" does not matter. 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 [17] (cooperative models), or to view a fuzzy system as a special neural network, and directly apply a learning algorithm [12] (hybrid models).

Approaches where neural networks are used to provide inputs for a fuzzy system, or to change the output of a fuzzy system, we prefer to call "neural (network)/fuzzy(system) combinations" or "concurrent neural/fuzzy models" to stress the difference that in these approaches parameters of a fuzzy system are not changed by a learning process. If the creation of a neural network is the main target, it is possible to apply fuzzy techniques to speed up the learning process, or to fuzzify a neural network by the extension principle to be able to process fuzzy inputs. These approaches could be called "fuzzy neural networks" to stress that fuzzy techniques are used to create or enhance neural networks.

2 Neuro–Fuzzy Classification

Classification of data is an area of application where statistical methods, machine learning and neural networks are thoroughly examined and successfully used. It is also possible to use a fuzzy system for classification with rules like

if x_1 is μ_1 **and** x_2 is μ_2 **and** ... **and** x_n is μ_n **then** pattern $(x_1, x_2, \ldots, x_n)$ belongs to class i,

where the $\mu_1, \ldots, \mu_n$ are fuzzy sets.

What is the advantage of having another method for classification? A fuzzy classifier is not a replacement for the aforementioned methods yielding better results, but a different way of achieving the same goal. If a decision is made for a fuzzy classifier usually the following advantages are considered:

- vague knowledge can be used,
- the classifier is interpretable in form of linguistic rules,
- from an applicational view the classifier is easy to implement, to use and to understand.

The rule base of a fuzzy classifier that uses rules like mentioned above, represents an approximation of an (unknown) function $\varphi : \mathbb{R}^n \to \{0,1\}^m$ that represents the classification task where $\varphi(\mathbf{x}) = (c_1, \ldots, c_m)$ such that $c_i = 1$ and $c_j = 0$ ($j \in \{1, \ldots, m\}, j \neq i$), i.e. $\mathbf{x}$ belongs to class C_i. Because of the inferences process, the rule base actually does not approximate φ but the function $\varphi' : \mathbb{R}^n \to [0,1]^m$. We can obtain $\varphi(\mathbf{x})$ by $\varphi(\mathbf{x}) = \psi(\varphi'(\mathbf{x}))$, where ψ reflects the interpretation of the classification result obtained from the fuzzy classifier. Usually the class with the largest activation value is chosen.

Classifiers are usually derived from data and are not specified directly. In case of a fuzzy classifier there two common methods:

(i) fuzzy clustering, and

(ii) neuro–fuzzy learning.

In the case of fuzzy clustering the input space is searched for clusters. The number of clusters is determined by an evaluation measure, and the size and shape of the clusters is given by the cluster algorithm. The obtained clusters can function as a fuzzy classifier, in fact, there is no need to express the classifier by rules. However, the interpretability is lost, and therefore fuzzy rules are sometimes created by projection of clusters (see Fig. 1).

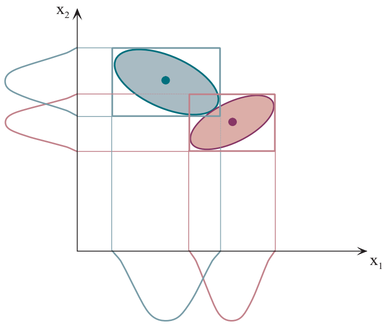

Figure 1: Projection of two fuzzy clusters to obtain fuzzy rules

Fuzzy rules obtained by projection of cluster suffer from a loss of information, as can be seen in Fig. 1. A fuzzy rule does not represent the cluster exactly, but only the smallest encompassing hyperbox. The performance of a fuzzy classifier obtained by clustering is therefore usually reduced, once it is expressed in form of fuzzy rules. In addition the rules are often hard to interpret, because the resulting fuzzy sets can have almost any shape.

Another method to obtain a fuzzy classifier from data is to use a neuro–fuzzy approach. This means the classifier is created from data by a heuristic learning procedure. If the neuro–fuzzy approach meets the five points listed in Section 1, then the interpretability of the resulting classifier, and a acceptable performance might be obtained. A neuro–fuzzy approach is also often computationally less expensive than a clustering approach, because of its simplicity.

A neuro–fuzzy classifier is nothing more than a fuzzy classifier obtained by a learning procedure, as described in Section 1. There are several known approaches to find a fuzzy classifier this way, e.g. FuNE [7], Fuzzy RuleNet [21], or NEFCLASS [11]. After creation of the classifier there is usually no hint on how it was derived. Therefore the term "neuro–fuzzy" strictly only applies to the creation or training phase. Afterwards, once the classifier is applied and not changed further, there remains a simple fuzzy system for classification. However the term "neuro–fuzzy classifier" is usually kept, to stress the mode of obtaining the classifier. The interpretation of the classifier is also often illustrated by representing it in a neural network structure like e.g. NEFCLASS in Fig. 2.

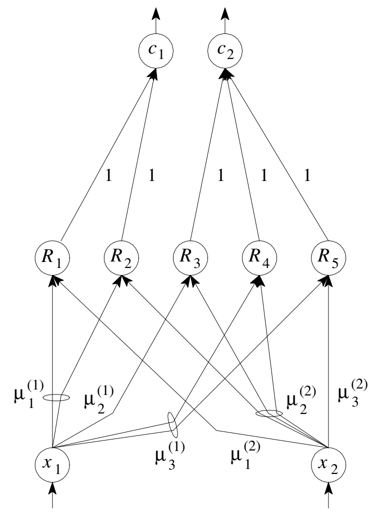

Figure 2: A NEFCLASS system represented as a 3–layer feedforward neural network

In the following section, we will present some new ideas for a learning algorithm to derive a fuzzy classifier from data, and illustrate our understaning of neuro–fuzzy learning.

3 Neuro–Fuzzy Learning

We have presented our NEFCLASS model (neuro–fuzzy classification) in [11], and some refined learning algorithms in [14]. Details about the model can be found there. Here we just want to illustrate the learning process by Figures 3 and 4.

The rule learning algorithm needs an initial fuzzy partitioning for each variable. This is e.g. given by a fixed number of equally distributed triangular membership functions (see Fig. 3). The combination of the fuzzy sets forms a "grid" in the data space, i.e. equally distributed overlapping rectangular clusters. Then the training data is processed, and those clusters that cover areas where data is located are added as rules into the rule base of the classifier. In a next step this rule base is reduced by just keeping the best performing rules [14]. The result after this stage of training can e.g. look like the situation in Fig. 3.

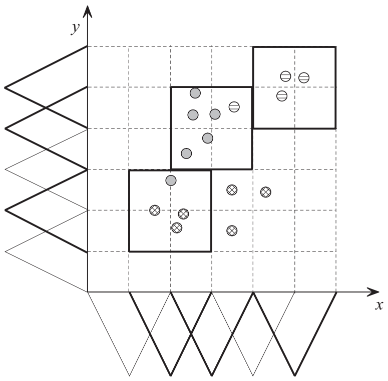

Figure 3: Situation after 3 fuzzy classification rules have been created using initial membership functions

After the rulebase has been created, the membership functions are tuned by a simple heuristic. For each rule a classification error is determined, and used to modify that membership function that is responsible for the rule activation (i.e. delivers the minimal membership degree of all fuzzy sets in the rules antecedent). The modification results in shifting the fuzzy set, and enlarging or reducing its support, such that a larger or smaller membership degree is obtained depending on the current error. The learning result might look like the situation in Fig. 4.

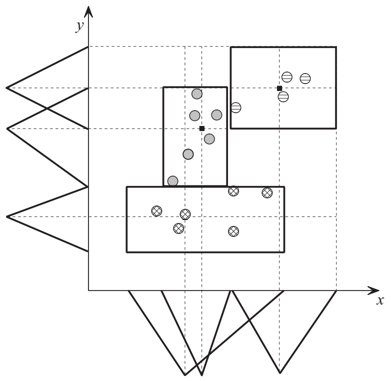

Figure 4: Situation after training the classifier, i.e. modifying the membership functions

To obtain an interpretable classifier some restrictions can be specified by the user. The NEFCLASS software [16] allows to impose the follwing restrictions on the learning algorithm:

- a membership function must not pass on of its neighbors,
- a membership function may be asymetrical,
- membership functions must intersect at 0.5.

How these restrictions influence the learning procedure is described in the following section.

To further enhance the learning capabilities, and to obtain classifiers that can be interpreted more easily, we introduce some new concepts for the NEFCLASS learning algorithm:

- Re–learning of the rule base: If the error cannot be further reduced by modifying the membership functions, the rule base can be learned anew. The "grid" given by the fuzzy sets is "distorted" during learning. It can be possible that restarting the creation and evaluation of the rule base leads to new and better rules, not considered before. The current rules are used as prior knowledge and are compared to newly created rules (if any).
- Rule pruning: To reduce the rule base, rule with poor performance, and rules that cover data that is also covered by other rules, can be deleted.

- Variable pruning: For each rule it is checked whether there are variables which never (or rarely) supply the minimal membership degree of all variables of the rule's antecedent. These variables can be deleted from the antecedent of the considered rule.

The impact of these additions to the learning process are discussed in a companion paper on this conference [15]. In the following section we present an example that shows the capabilities of our neuro–fuzzy learning strategy.

4 NEFCLASS – An Example

As an example for the learning capabilities of NEFCLASS we use the "Wisconsin Breast Cancer" data set [23]. This data set contains 699 cases distributed into two classes (benign and malign). We used only 683 cases (342 for training, 341 for testing), because 16 cases have missing values. Each pattern has nine features.

To show how NEFCLASS performs when prior knowledge is supplied, we used a fuzzy clustering method to obtain fuzzy rules. We used a modification of the algorithm by Gustafson and Kessel [6] presented in [8]. Fuzzy clustering discovered three clusters that were interpreted as fuzzy rules by projecting the clusters to each dimension and finding trapezoidal membership functions that closely matched the projections. The membership functions were interpreted by *small, medium* and *large* resulting in three rules that caused 94 classification errors on the whole data set:

R_1: **if** (s,s,s,s,s,s,s,s,s) **then** *benign*,
R_2: **if** (m,m,m,m,m,l,m,m,s) **then** *malign*,
R_3: **if** (m,l,l,l,m,l,m,l,s) **then** *malign*.

Feature x_9 has the same value in all rules, therefore we can leave it out in the NEFCLASS system.

When we initialize a NEFCLASS system with these three rules, we at first get 240 classification errors. However, after 80 epochs of training, using the constraint, that fuzzy sets must not pass each other, we obtained a result of only 50 errors altogether (92.7% correct).

A neuro–fuzzy learning strategy is a tool to support the creation of a fuzzy classifier, but not to completely automate it. This means the user should supervise the learning process, and interpret the result. By analyzing the fuzzy sets obtained by training the three rules used as prior knowledge we found that the fuzzy set *medium* substantially overlapped with either the fuzzy set *small* or *large* for almost all variables. This can be seen as evidence that *medium* is superfluous.

Therefore we again trained a NEFCLASS system with "best per class" rule learning, allowing it to create four rules (two for each class). This time we only used two fuzzy sets to partition the domains of each variable. After 100 epochs of training NEFCLASS made only 24 errors on the complete set (96.5% correct). It has found the rules

R_1: **if** (s,s,s,s,s,s,s,s,s) **then** *benign*,
R_2: **if** (l,s,s,s,s,s,s,s,s) **then** *benign*,
R_3: **if** (l,l,l,l,l,l,l,l,s) **then** *malign*,
R_4: **if** (l,l,l,l,s,l,l,l,s) **then** *malign*.

If we additionally apply the constraint that the fuzzy sets must intersect at membership degree 0.5 we obtain 26 classification errors on the whole set.

5 Conclusions

We have discussed our notion of a neuro–fuzzy system in the context of fuzzy classification. We consider a neuro–fuzzy method to be a tool for creating fuzzy systems from data. The learning algorithm should take the semantics of the desired fuzzy system into account, and adhere to certain constraints. The learning result should also be interpreted, and the insights gained by this should be used to restart the learning procedure to obtain better results if necessary.

A fuzzy classifier, especially a neuro–fuzzy classifier, is only used, when interpretation, and the employment of (vague) prior knowledge is required. Fuzzy classification is not a replacement, but an addition to other methods like statistics or neural networks. The price for the interpretation of the classifier in form of simple fuzzy rules, and for the simple and computationally efficient learning algorithm might be paid by a classification result that is not as good as it could be if other methods are used.

The example with the Wisconsin Breast Cancer data shows that NEFCLASS can be used as an interactive data analysis method. It is useful to provide prior knowledge when it is possible. A combination with fuzzy clustering can help here. The learning result of NEFCLASS should be analyzed, and the obtained information can be used for another run that yields an even better result.

References

[1] Hamid R. Berenji and Pratap Khedkar. Learning and tuning fuzzy logic controllers through reinforcements. *IEEE Trans. Neural Networks*, 3:724–740, September 1992.

[2] James C. Bezdek, Eric Chen-Kuo Tsao, and Nikhil R. Pal. Fuzzy Kohonen clustering networks. In *Proc. IEEE Int. Conf. on Fuzzy Systems 1992*, pages 1035–1043, San Diego, CA, 1992.

[3] James J. Buckley and Yoichi Hayashi. Fuzzy neural networks: A survey. *Fuzzy Sets and Systems*, 66:1–13, 1994.

[4] James J. Buckley and Yoichi Hayashi. Neural networks for fuzzy systems. *Fuzzy Sets and Systems*, 71:265–276, 1995.

[5] Dimiter Driankov, Hans Hellendoorn, and Michael Reinfrank. *An Introduction to Fuzzy Control*. Springer-Verlag, Berlin, 1993.

[6] D.E. Gustafson and W.C. Kessel. Fuzzy clustering with a fuzzy covariance matrix. In *Proc. IEEE CDC*, pages 761–766, San Diego, CA, 1979.

[7] Saman K. Halgamuge and Manfred Glesner. Neural networks in designing fuzzy systems for real world applications. *Fuzzy Sets and Systems*, 65:1–12, 1994.

[8] Frank Klawonn and Rudolf Kruse. Constructing a fuzzy controller from data. *Fuzzy Sets and Systems*, 85:177–193, 1997.

[9] Rudolf Kruse, Jörg Gebhardt, and Frank Klawonn. *Foundations of Fuzzy Systems*. Wiley, Chichester, 1994.

[10] Detlef Nauck, Frank Klawonn, and Rudolf Kruse. *Foundations of Neuro-Fuzzy Systems*. Wiley, Chichester, 1997.

[11] Detlef Nauck and Rudolf Kruse. NEFCLASS – a neuro-fuzzy approach for the classification of data. In K. M. George, Janice H. Carrol, Ed Deaton, Dave Oppenheim, and Jim Hightower, editors, *Applied Computing 1995. Proc. 1995 ACM Symposium on Applied Computing, Nashville, Feb. 26–28*, pages 461–465. ACM Press, New York, February 1995.

[12] Detlef Nauck and Rudolf Kruse. Designing neuro-fuzzy systems through backpropagation. In Witold Pedrycz, editor, *Fuzzy Modelling: Paradigms and Practice*, pages 203–228. Kluwer, Boston, 1996.

[13] Detlef Nauck and Rudolf Kruse. Neuro-fuzzy systems research and applications outside of Japan (in japanese). In M. Umano, I. Hayashi, and T. Furuhashi, editors, *Fuzzy-Neural Networks* (in Japanese), Soft Computing Series, pages 108–134. Asakura Publ., Tokyo, 1996.

[14] Detlef Nauck and Rudolf Kruse. A neuro-fuzzy method to learn fuzzy classification rules from data. *Fuzzy Sets and Systems*, 89:277–288, 1997.

[15] Detlef Nauck and Rudolf Kruse. New learning strategies for NEFCLASS. In *Proc. Seventh International Fuzzy Systems Association World Congress IFSA'97*, volume IV, pages 50–55, Prague, 1997.

[16] Detlef Nauck, Ulrike Nauck, and Rudolf Kruse. Generating classification rules with the neuro-fuzzy system NEFCLASS. In *Proc. Biennial Conference of the North American Fuzzy Information Processing Society NAFIPS'96*, Berkeley, CA, June 1996.

[17] Witold Pedrycz and H. C. Card. Linguistic interpretation of self-organizing maps. In *Proc. IEEE Int. Conf. on Fuzzy Systems 1992*, pages 371–378, San Diego, CA, 1992.

[18] T. J. Procyk and E. H. Mamdani. A linguistic self-organizing process controller. *Automatica*, 15:15–30, 1979.

[19] Wu Zhi Qiao, Wang Pei Zhuang, Teh Hoon Heng, and Song Shou Shan. A rule self-regulating fuzzy controller. *Fuzzy Sets and Systems*, 47:13–21, 1992.

[20] Shihuang Shao. Fuzzy self-organizing controller and its application for dynamic processes. *Fuzzy Sets and Systems*, 26:151–164, 1988.

[21] Nadine Tschichold-Gürman. Generation and improvement of fuzzy classifiers with incremental learning using fuzzy rulenet. In K. M. George, Janice H. Carrol, Ed Deaton, Dave Oppenheim, and Jim Hightower, editors, *Applied Computing 1995. Proc. 1995 ACM Symposium on Applied Computing, Nashville, Feb. 26–28*, pages 466–470. ACM Press, New York, February 1995.

[22] Petri Vuorimaa. Fuzzy self-organizing map. *Fuzzy Sets and Systems*, 66:223–231, 1994.

[23] W.H. Wolberg and O.L. Mangasarian. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proc. National Academy of Sciences*, 87:9193–9196, December 1990.