Neuro–Fuzzy Classification with NEFCLASS

Detlef Nauck and Rudolf Kruse

Technical University of Braunschweig, Dept. of Computer Science,
Bueltenweg 74–75, D–38106 Braunschweig, Germany
Tel: +49.531.391.3155, Fax: +49.531.391.5936, E-Mail: D.Nauck@tu-bs.de

1 Introduction

In this paper we present an approach to neuro–fuzzy data analysis. The goal is to derive fuzzy rules from a set of data that can be separated in different crisp classes. The fuzzy rules describing the data are of the form:

$$R: \textbf{if } x_1 \text{ is } \mu_1 \textbf{ and } x_2 \text{ is } \mu_2 \textbf{ and } \ldots \textbf{ and } x_n \text{ is } \mu_n \\ \textbf{then the pattern } (x_1, x_2, \ldots, x_n) \text{ belongs to class } i,$$

where $\mu_1, \ldots, \mu_n$ are fuzzy sets. The task of the NEFCLASS model (**NE**uro **F**uzzy **CLASS**ification) is to discover these rules and to learn the shape of the membership functions.

NEFCLASS is derived from a generic fuzzy perceptron [8, 12]. It has the structure of a 3–layer feedforward perceptron, and uses fuzzy sets as wheigts on the connections from the input to the hidden layer. The connections from the hidden to the output layer are fixed at unity. The input units store the input features, the hidden units (rule nodes) represent fuzzy rules, and the output units represent the pattern classes.

2 The NEFCLASS model

NEFCLASS is used to determine the correct class or category of a given input pattern. The patterns are vectors $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ and a class C is a (crisp) subset of $\mathbb{R}^n$. We assume the intersection of two different classes to be empty. The pattern feature values are represented by fuzzy sets, and the classification is described by a set of linguistic rules. For each input feature x_i there are q_i fuzzy sets $\mu_1^{(i)}, \ldots, \mu_{q_i}^{(i)}$, and the rule base contains k fuzzy rules $R_1, \ldots, R_k$.

The rule base is an approximation of an (unknown) function $\varphi : \mathbb{R}^n \rightarrow \{0,1\}^m$ 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

This paper appears in: Kleinschmidt, P. et al: Operations Research Proceedings 1995, Springer, Berlin, 1996: pp 294–299.

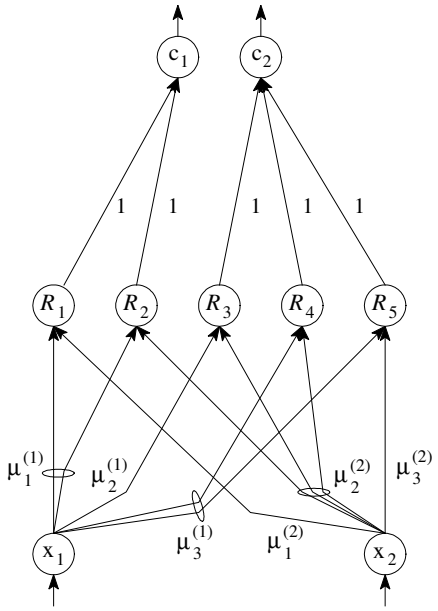

Fig. 1. A NEFCLASS system with two inputs, five rules and two output classes

of the mathematics involved the rule base actually does not approximate φ but the function $\varphi' : \mathbb{R}^n \to [0,1]^m$. We will get $\varphi(\mathbf{x})$ by $\varphi(\mathbf{x}) = \psi(\varphi'(\mathbf{x}))$, where ψ reflects the interpretation of the classification result obtained from a NEFCLASS system. In our case we will map the highest component of each vector $\mathbf{c}$ to 1 and its other components to 0, respectively.

The fuzzy sets and the linguistic rules which perform this approximation, and define the resulting NEFCLASS system, will be obtained from a set of examples by learning. The Fig. 1 shows a NEFCLASS system that classifies input patterns with two features into two distinct classes by using five linguistic rules.

Its main feature are the *shared weights* on some of the connections. This way we make sure, that for each linguistic value (e.g. "x_1 is *positive big*") there is only one representation as a fuzzy set (e.g. $\mu_1^{(1)}$ in Fig. 1), i.e. the linguistic value has only one interpretation for all rule units (e.g. R_1 and R_2 in Fig. 1). It cannot happen that two fuzzy sets that are identical at the beginning of the learning process develop differently, and so the semantics of the rule base encoded in the structure of the network is not affected [10]. Connections that share a weight always come from the same input unit.

A NEFCLASS system can be build from partial knowledge about the patterns, and can be then refined by learning, or it can be created from scratch by learning. A user has to define a number of initial fuzzy sets partitioning the domains of the input features, and must specify the largest number of rule nodes that may be created in the hidden layer.

Each fuzzy set $\mu_j^{(i)}$ is labelled with a linguistic term $A_j^{(i)}$. This may be terms like *small, medium, large* etc. The fuzzy sets of those connections leading to the same rule unit R are also called the *antecedents* of R.

A NEFCLASS system that is created from scratch starts with no hidden units at all. They are created during a first run through the learning task $\tilde{\mathcal{L}}$, i.e. the set of examples. A rule is created by finding for a given input pattern $\mathbf{p}$ the combination of fuzzy sets, where each yields the highest degree of membership for the respective input feature. If this combination is not identical to the antecedents of an already existing rule, and if the permitted number of rule units is not yet reached, a new rule node is created. This is a very simple way of finding fuzzy rules, and it can only be successful, if the patterns are selected randomly from the learning task, and if the cardinalities of the classes are approximately equal. Another way is to create as much rules as can be found within the data set, then use a scoring system and then select those rules that have the highest scores. Both methods have been implemented in the NEFCLASS software.

When the rule base is created, the learning algorithm will adapt the fuzzy sets of the antecedents. We use triangular membership functions, and min as the t–norm to determine the degree of fulfillment of a rule, i.e. the activation of a rule node. The adaption of the membership functions is done by a supervised learning procedure. For each output unit the error is determined, and propagated back through the network. Using a reward/punishment scheme, the membership functions are shifted, and their support is scaled, such that the performance of the rules is enhanced. For the full description of the learning algorithm please see [11].

3 Deriving Fuzzy Rules From Data

In this section we will present the results of a NEFCLASS system applied to a data set. The task is to obtain classification rules to correctly classify the patterns of the well known Iris data set [1]. This data set contains 150 patterns belonging to three different classes (Iris Setosa, Iris Versicolour, and Iris Virginica) with 50 patterns each. The patterns have four input features (sepal and petal length and width of the iris flower). The first class is linearly separable from the other two classes, whereas the second and third class are not linearly separable from each other. This is a very simple domain, and it is often used as a benchmark.

For the NEFCON learning process the data set was split in half, and the patterns were ordered alternately within the training and test data sets. We allowed the system to create a maximum of seven rules using the simple rule learning algorithm. The domains of the four input variables were each partitioned by three initially equally distributed fuzzy sets. The rules was created during the first sweep through the training data set, after that the membership functions were trained for 500 epochs.

After learning, 3 out of 75 patterns from the training set were still classified wrongly (i.e. 96% correct). Testing the second data set, the NEFCLASS system classified only 2 out 75 patterns incorrectly (i.e. 97.3% correct). Considering all 150 patterns the system performed well with 96.67% correct classifications.

Table 1. The rules to classify the Iris data set learned by NEFCLASS

IF V_1 is small	and V_2 is medium	and V_3 is small	and V_4 is small	THEN Cl. 1
IF V_1 is large	and V_2 is medium	and V_3 is medium	and V_4 is medium	THEN Cl. 2
IF V_1 is small	and V_2 is small	and V_3 is medium	and V_4 is medium	THEN Cl. 2
IF V_1 is medium	and V_2 is medium	and V_3 is large	and V_4 is large	THEN Cl. 3
IF V_1 is medium	and V_2 is small	and V_3 is large	and V_4 is large	THEN Cl. 3
IF V_1 is large	and V_2 is small	and V_3 is large	and V_4 is large	THEN Cl. 3
IF V_1 is large	and V_2 is medium	and V_3 is large	and V_4 is large	THEN Cl. 3

In Table 1 the fuzzy rules learned by NEFCLASS are shown. NEFCLASS started with no rule in the beginning, and created one rule to manage the first and easiest class. Two classify the patterns of the second and third class, two and four rules were created.

Fig. 2. The fuzzy sets for the Iris data set learned by NEFCLASS

In Fig. 2 the fuzzy sets learned by NEFCLASS are shown. The system is initialized by three equally distributed fuzzy sets (shouldered or triangular) for each input feature. The learning result shows, that the fuzzy sets labeled with *medium* are obviously superfluous for the first two variables, and a partitioning of the domains into *small* and *large* would be sufficient. The membership function

of the third and fourth variable have been corrected only a little bit.

We have compared the learning result of NEFCLASS to the results obtained by a multilayer perceptron trained with backpropagation, by another neuro–fuzzy approach called FuNe [3], and by a fuzzy clustering procedure.

For the neural network approach we used a multilayer perceptron of 4 input, 3 hidden and 3 output units. The network was able to learn the training set completely, i.e. there were no classification errors on the 75 training patterns. But the network classified 5 of the 75 test patterns incorrectly, using the same criterion for classification as we did for the NEFCLASS system. So considering all 150 patterns both models performed equally well. Incrementing the number of hidden units within the neural network caused more training cycles, and ususally did not produce a better result. However, we found one network with 10 hidden units that classified only 3 patterns of the test set incorrectly. Of course the learning results of a neural network are in no way interpretable.

Another neuro–fuzzy approach to pattern classification is presented in [3]. The authors use a system called "FuNe I" to obtain rules and fuzzy sets from data samples. The authors of [3] have applied their FuNe I system to several real world examples, and also used the IRIS data set split in a training and a test set. They have reached a classification rate of 99% on the test set using 13 rules and four inputs, and a 96% classification rate using 7 rules and 3 inputs. FuNe I is offered in a limited test version by the authors of [3], so we could run our own test to compare it with NEFCLASS. We allowed the system to create 10 rules, and it came out with 5 classification errors on the test set after training. Therefore the two models are comparable in their performance, even though FuNe I has a much more complex structure and training procedure. FuNe I also uses a concept of weighted rules. We refrained from using this approach in NEFCLASS, because the semantics of weighted fuzzy rules is not clear [7, 9].

As a fuzzy clustering method we used variation of the Gath and Geva algorithm [2] that was developed in [4]. For the clustering process all 150 patterns were used together, and we obtained 8 fuzzy rules from the clustering procedure resulting in 7 errors (i.e. 95.3% correct) on the whole data set: 3 misclassifications, and 4 patterns that are not classified at all due to loss of information caused by process of rule creation, where membership functions have to be approximated.

Even though a sophisticated fuzzy clustering procedure usually produces better classification results as a NEFCLASS system [5], this advantage is reduced, once the derivation of interpretable fuzzy rules is of interest. By approximating fuzzy sets a loss of information may occur, resulting in patterns that are not classified by any of the rules, and the classification performance becomes worse. The resulting fuzzy rules are also often not as easy to interpret as the fuzzy rules created by NEFCLASS. The advantage of a NEFCLASS system is that it enhances its performance by directly manipulating given initial fuzzy partitionings. It is easily possible to define constraints for the learning algorithm like "fuzzy sets may not pass each other", "the support of a fuzzy set may not collapse to a singleton", etc. This way the learning result is more likely to be interpretable, and the semantics of the linguistic terms assigned to the fuzzy sets is conserved.

4 Conclusions

In this paper we have presented the NEFCLASS model, a neuro–fuzzy classification system. The model is able to create fuzzy rules from data examples by a single run through the data set. After creating a rule base NEFCLASS learns its fuzzy sets by adapting parameters of the membership functions in a supervised learning algorithm.

NEFCLASS was tested on the IRIS data set and the performance was satisfactory compared to usual neural networks, another neuro–fuzzy approach, and a fuzzy clustering approach.

The NEFCLASS software for MS–DOS PC can be freely obtained by anonymous ftp from ftp.ibr.cs.tu-bs.de in the directory /pub/local/nefclass, or from the World Wide Web (http://www.cs.tu-bs.de/~nauck).

References

1. R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annual Eugenics*, 7(Part II):179–188, 1936.
2. I. Gath and A.B. Geva. Unsupervised optimal fuzzy clustering. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11:773–781, 1989.
3. Saman K. Halgamuge and Manfred Glesner. Neural networks in designing fuzzy systems for real world applications. *Fuzzy Sets and Systems*, 65:1–12, 1994.
4. Frank Klawonn and Rudolf Kruse. Constructing a fuzzy controller from data. *Fuzzy Sets and Systems*, 85(1), 1997.
5. Frank Klawonn, Detlef Nauck, and Rudolf Kruse. Generating rules from data by fuzzy and neuro–fuzzy methods. In *Proc. Fuzzy–Neuro–Systeme'95*, pages 223–230, Darmstadt, November 1995.
6. Rudolf Kruse, Jörg Gebhardt, and Rainer Palm, editors. *Fuzzy Systems in Computer Science*. Vieweg, Braunschweig, 1994.
7. Detlef Nauck. Fuzzy neuro systems: An overview. In [6], pages 91–107. Vieweg, Braunschweig, 1994.
8. Detlef Nauck. A fuzzy perceptron as a generic model for neuro–fuzzy approaches. In *Proc. Fuzzy–Systeme'94*, Munich, October 1994.
9. Detlef Nauck, Frank Klawonn, and Rudolf Kruse. *Neuronale Netze und Fuzzy–Systeme*. Vieweg, Wiesbaden, 1994.
10. Detlef Nauck and Rudolf Kruse. Interpreting changes in the fuzzy sets of a self-adaptive neural fuzzy controller. In *Proc. Second Int. Workshop on Industrial Applications of Fuzzy Control and Intelligent Systems (IFIS'92)*, pages 146–152, College Station, Texas, December 1992.
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. of the 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.