A Neuro-Fuzzy Classifier and Its Applications

Chuen-Tsai Sun
Department of Computer and Information Science
National Chiao Tung University
Hsinchu, Taiwan 30050
E-mail: ctsun@weber.cis.nctu.edu.tw

Jyh-Shing Jang
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA 94720
E-mail: jang@diva.berkeley.edu

Abstract

Fuzzy classification is the task of partitioning a feature space into fuzzy classes. A learn-by-example mechanism is desirable to automate the construction process of a fuzzy classifier. In this paper we introduce a method of employing *adaptive networks* to solve a fuzzy classification problem. System parameters, such as the *membership functions* defined for each feature and the *parameterized t-norms* used to combine conjunctive conditions are calibrated with backpropagation.

To explain this new approach, first we introduce the concept of adaptive networks and derive a supervised learning procedure based on a gradient descent algorithm to update the parameters in an adaptive network. Next, we apply the proposed architecture to two problems: two-spiral classification and Iris categorization. From the experiment results, it is summarized that the adaptively adjusted classifier performs well on an Iris classification problem. The results are discussed from the viewpoint of feature selection.

I. Introduction

Conventional approaches of pattern classification involve clustering training samples and associating clusters to given categories. The complexity and limitations of previous mechanisms are largely due to the lacking of an effective way of defining the boundaries among clusters. This problem becomes more intractable when the number of features used for classification increases.

On the contrary, *fuzzy classification* [9, 14] assumes the boundary between two neighboring classes as a continuous, overlapping area within which an object has partial membership in each class. This viewpoint not only reflects the reality of many applications in which categories have fuzzy boundaries, but also provides a simple representation of the potentially complex partition of the feature space. In brief, we use *fuzzy if-then rules* to describe a classifier. A typical fuzzy classification rule is like:

$$\text{if } X_1 \text{ is } A \text{ and } X_2 \text{ is } B \text{ then } Z \text{ is } C,$$

where X_1 and X_2 are features or input variables; A, B are *linguistic terms* [13] characterized by appropriate *membership functions* [12], which describe the features of an object Z. The *firing strength* or the degree of appropriateness of this rule with respect to a given object is the degree of belonging of this object to the class C.

As such, a fuzzy rule gives a meaningful expression of the qualitative aspects of human recognition. Based on the result of pattern matching between rule antecedents and input signals, a number of fuzzy rules are triggered in parallel with various values of firing strength. Individually invoked actions are considered together with a combination logic.

Further, we want the system to have learning ability of updating and fine-tuning itself based on newly coming information. Researchers have been trying to automate the classifier construction process based on a training data set. We propose a method of using adaptive networks for this purpose. We use experimental data to verify the effectiveness of this approach.

II. Learning with Adaptive Networks

An adaptive network is a multi-layer feed-forward network in which each node performs a particular function (*node*

0-7803-0614-7/93$03.00 ©1993IEEE

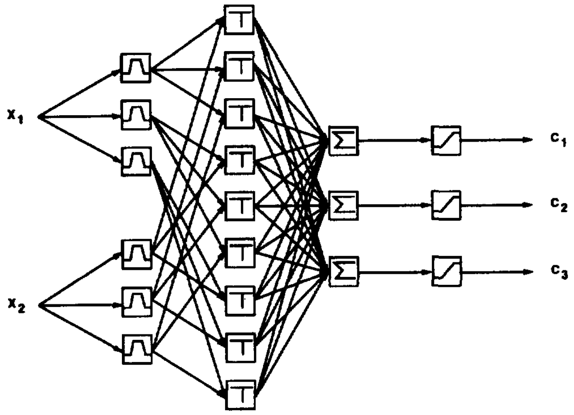

Figure 1: *An adaptive-network-based fuzzy classifier.*

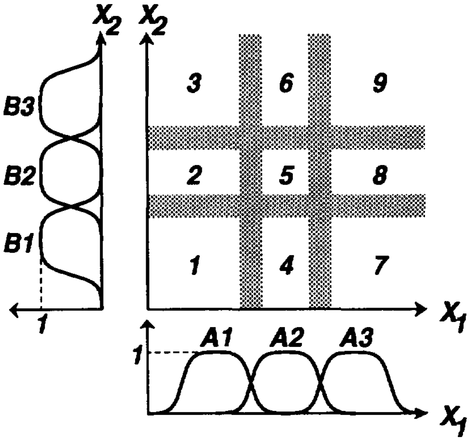

Figure 2: *Partition of feature space.*

function) based on incoming signals and a set of parameters pertaining to this node. The type of node function may vary from node to node; and the choice of node function depends on the overall function that the network is designed to carry out.

Figure 1 demonstrates the adaptive-network-based classifier architecture with two input variables, X_1 and X_2. The training data are categorized by three classes, C_1, C_2 and C_3. Each input is represented as three linguistic terms, thus we have nine fuzzy rules. In our model the nodes in the same layer have the same type of node function.

Each node in *Layer 1* is associated with a parameterized bell-shaped membership function represented as

$$\mu_A(X_i) = \frac{1}{1 + [(\frac{x_i - c_i}{a_i})^2]^{b_i}}, \qquad (1)$$

where X_i is one of the input variables, A is the linguistic term associated with this node function, and $\{a_i,\ b_i,\ c_i\}$ is the parameter set.

The initial values of the parameters are set in such a way that the membership functions along each axis satisfy ϵ ***completeness*** [6] ($\epsilon = 0.5$ in our case), ***normality*** and ***convexity*** [5]. Figure 2 illustrates the concept. Although these initial membership functions are set heuristically and subjectively, they do provide an easy interpretation parallel to human thinking. The parameters are then tuned with ***backpropagation***, a gradient descent method, in the learning process based on a given training data set.

Each node in *Layer 2* generates a signal corresponding to the conjunctive combination of individual degrees of match. The output signal corresponds to the firing strength of a fuzzy rule with respect to an object to be categorized. In most pattern classification and query-retrieval systems, the conjunction operator plays an important role and its interpretation changes across contexts. Since there does not exist a single operator that is suitable for all applications, we can use ***parameterized t-norms*** at *Layer 2* to cope with this dynamic property of classifier design. Bonissone provided a detailed discussion on t-norms and their parameterized versions, see [2]. For example, we can use Hamacher's t-norm:

$$T_H(x_1, x_2, \gamma) = \frac{x_1 x_2}{\gamma + (1-\gamma)(x_1 + x_2 - x_1 x_2)}, \qquad (2)$$

where x_i's are the operands and γ is a non-negative parameter.

In some other applications, e.g., see [15], features are combined in a compensatory way. For these situations, ***mean operators*** [11] are more appropriate than conjunctive operators. To find a good mean operator for a certain system, we can also implement a parameterized operator and use training data to calibrate it. For instance, we can use the one proposed by Dyckhoff and Pedrycz:

$$M_{DP}(x_1, x_2, \gamma) = \frac{(x_1^\gamma + x_2^\gamma)^{1/\gamma}}{2}, \qquad (3)$$

where $\gamma \geq 1$. Note that we can either use a parameterized operator for each node in *Layer 2* or employ a single one for the whole layer. Whether an operator is local or global depends on applications. Moreover, a ***parameterized fuzzy quantifier*** [3] can also be introduced into this picture based on the same concept. The combinational parameters are also fine-tuned by backpropagation.

We take the linear combination of the firing strengths of the rules at *Layer 3* and apply a sigmoidal function at *Layer 4* to calculate a degree of belonging to a certain

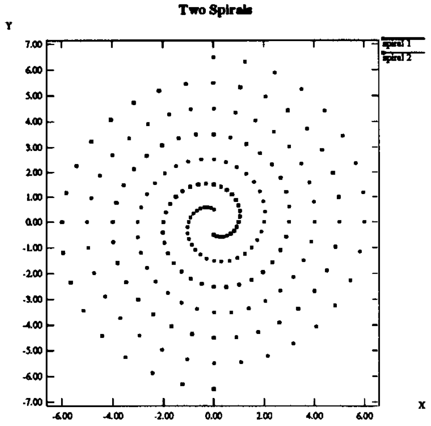

Figure 3: *Training data for the two-spiral problem.*

class. Through experience we found the following definition of error measure useful in classification problems. As before, assume we have three classes, the error measure E can be formulated as:

$$\begin{aligned} E \quad = \quad & do_1(1-do_2)(1-do_3)\{(sig[m(co_2-co_1)] \\ & \quad + sig[m(co_3-co_1)]\} \\ & + (1-do_1)do_2(1-do_3)\{(sig[m(co_1-co_2)] \\ & \quad + sig[m(co_3-co_2)]\} \\ & + (1-do_1)(1-do_2)do_3\{(sig[m(co_1-co_3)] \\ & \quad + sig[m(co_2-co_3)]\} \end{aligned} \tag{4}$$

where do_i's are desired outputs and co_i's are calculated outputs. The first, second and third terms account for the conditions when the desired classes are class 1, 2 and 3, respectively. Since this error measure is reasonable when the maximum selector is used, it is therefore referred to as ***maximum-type error measure***.

The maximum-type error measure introduced above can increase the degrees of freedom and therefore is suitable for crisp-output neuro-fuzzy classifiers. Meanwhile, the slow convergence of the gradient descent is compensated for by the proper choice of the maximum-type error measure, so the learning process will not suffer from the drawbacks of the gradient descent. In the following we present two application examples which employ neuro-fuzzy classifiers with maximum-type error measures to do crisp pattern classification. with maximum-type error measures to do crisp pattern classification.

III. Two-Spiral Problem

The two-spiral problem was proposed by Alexis P. Wieland on the connectionist mailing list as an interesting benchmark task for neural networks. The task requires a neural network classifier with two inputs and one output to learning a mapping that distinguishes between points of two intertwined spirals. The two sets of spiral data consist of 194 points, with 97 points for each spiral. One spiral is generated as a mirror image of the other, making the problem highly nonlinear-separable.

As pointed out by Wieland, this task has several features that makes it an interesting test for neural network's learning algorithms. First of all, it requires the neural networks to learn the highly nonlinear separation of the input space, which is difficult for most current learning algorithms. Secondly, its 2-dimensional input space makes it easy to plot the overall input-output relations as a 3-dimensional surface or 2-dimensional image for visual inspection and analysis.

To proceed with the simulation, first the rule number has to be decided. Since the input partition is checkerboard-like, we expect that the partition number (or equivalently, the number of membership functions) on either input x and y should be equal to the maximum number of alternations between classes along one dimension when the other is fixed. In the two-spiral problem, the maximum number of alternations on y is approximately 14, which occurs on the straight line $x = 0$; the maximum number of alternations on x is approximately 13, which occurs on $y = 0$.

Using the maximum-type error measure defined above, we perform four runs of simulation; the number of membership functions on both inputs is varied from 10 to 13 sequentially. It is found that 13 is the minimum number for the network to classify the two spirals correctly. This agrees with our observation of the maximum number of alternations along each dimension.

As mentioned earlier, this problem is suitable for visual inspection or analysis on the classifier's input-output behavior through data visualization techniques such as 3-D surface or 2-D image. Figure 4 depicts the classifier's input-output behavior; each of the four images is composed of 22,500 (150×150) pixels which are 2 bit deep.

Although we can always employ a large number of membership to achieve a perfect classification, this kind of over-parameterized structure is not recommended since it not only slows down the learning but also degrades the generalization power for unseen data sets; this is just like the case in over-parameterized neural networks. Therefore, the ability to determine the number of membership functions from visual inspection is a very practical and useful technique that enables us to find roughly a minimum structure of a neuro-fuzzy classifier to do the job. For neural networks, we do not have similar quick and easy techniques to determine the minimum structure (node numbers and layer numbers) simply due to the uniformity in the node function.

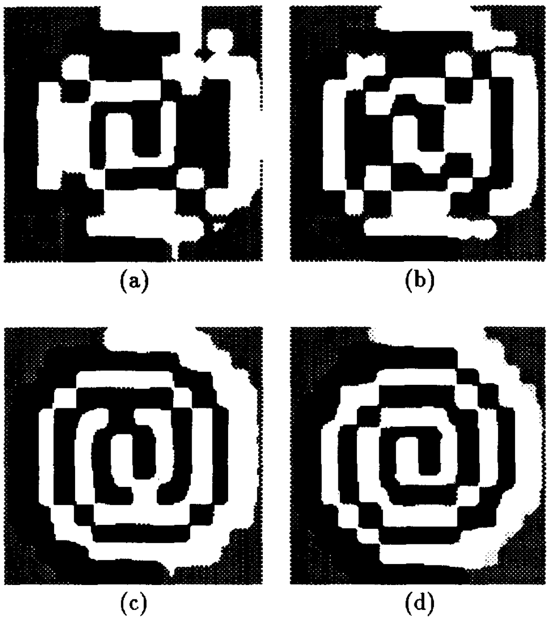

Figure 4: *Image representation of classifier's input-output behavior*. The number of membership functions on x and y equals to (a) 10; (b) 11; (c) 12 and (d) 13.

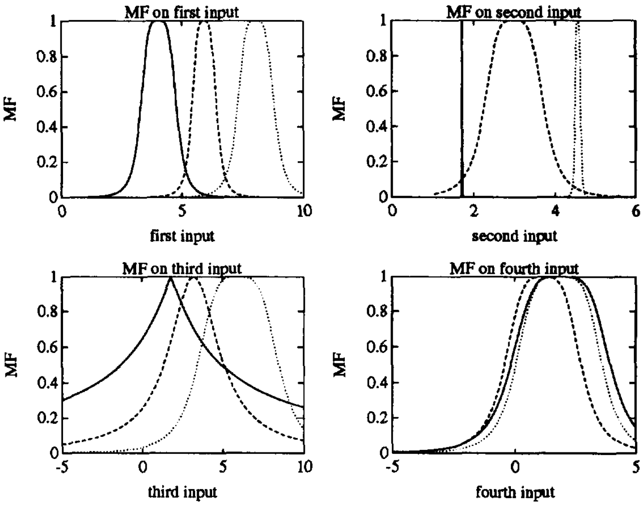

Figure 5: *Membership functions after 20 training epochs in Iris classification*.

IV. Iris Categorization

Next we apply the proposed scheme on an Iris classification problem of finding the mapping between four input variables (sepal length, sepal width, petal length, and petal width) and three classes (Setosa, Versicolor, and Virginica).

There are 150 samples in the data set and we use 120 of them as training data and the other 30 for testing. Initially, each feature dimension is partitioned into 3 homogeneously distributed overlapping regions. We constructed a network and trained it with the training data set for 20 epochs. The adjusted network was then evaluated by the testing data set. The desired and calculated outputs matched for all 30 testing data. The discriminating power of the classifier was well validated.

Another advantage of our fuzzy approach is that the resulted model gives us insights of the data characteristics. We analyzed the data set with statistical methods and found that, individually speaking, both inputs 3 and 4 have stronger (but incomplete) discriminating ability than inputs 1 or 2. However, if we choose input 3 as the primary salient feature, we need input 1 to be the secondary feature to complete the feature space partition. This analytical conclusion was predicted by the adjusted parameters in our adaptive network model. Figure 5 shows the final membership functions.

In this figure, the membership functions of the third input clearly gives the ranges of the salient feature. On the other hand, inputs 2 and 4 are neglected by different ways. For input 2, two of the initial functions shrank to peaks and left the remaining one to cover the entire dimension. For input 4, the three functions adjusted themselves to overlap each other and redundantly covered the dimension. The various behavior remains an interesting research topic to be further studied in the future.

V. Concluding remarks

An adaptive classifier partitions the feature space based on labeled training data. In the context of fuzzy classification, classes are overlapping and each training data item is associated with numbers in the unit interval representing degrees of belonging, one value for each class. The overlapping among regions provides the natural smoothness for the input-output mapping. This characteristic makes this model suitable for classification problems, especially for those with overlapping categories.

We proposed a general fuzzy classification scheme with learning ability using an adaptive network, which can be used in pattern recognition, decision analysis, and many other fields. Membership parameters were identified with the model. Parameterized t-norms and mean operators were brought into this picture to make the classification scheme more flexible. The resulted membership function served the need of feature selection.

The proposed neuro-fuzzy approach is better than neural network classifiers in the sense that prior knowledge about the training data set can be encoded into the parameters of the neuro-fuzzy classifier. This encoded knowledge, usually acquired from human experts or data visualization techniques, can almost always allow the learning process to begin from a good initial point not far away from the optimal one in the parameter space, thus speeding up the convergence to the optimal or a near-optimal point. Moreover, the parameters obtained after the learning process can be easily transformed into structure knowledge in the form of fuzzy if-then rules.

References

[1] K. J. Astrom and B. Wittenmark. *Computer Controller Systems: Theory and Design.* Prentice-Hall, Inc., 1984.

[2] Piero P. Bonissone. Summarizing and propagating uncertain information with triangular norms. *International Journal of Approximate Reasoning*, 1:71–101, 1987.

[3] J. Kacprzyk and R. R. Yager. "Softer" optimization and control models via fuzzy linguistic quantifiers. *Information Sciences*, 34:157–178, 1984.

[4] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME. Journal of Basic Engineering*, pages 35–45, March 1960.

[5] Arnold Kaufmann and Madan M. Gupta. *Introduction to Fuzzy Arithmetic: Theory and Applications.* Van Nostrand Reinhold Co., 1985.

[6] C.C. Lee. Fuzzy logic in control systems: Fuzzy logic controller. *IEEE Trans. on Systems, Man, and Cybernetics*, 20(2):404–435, 1990.

[7] John Moody and Christian Darken. Learning with localized receptive fields. Technical Report YALEU/DCS/RR-649, Department of Computer Science, Yale University, 1988.

[8] Stephen M. Omohundro. Geometric learning algorithms. Technical Report TR-89-041, International Computer Science Institute, 1989.

[9] Sergei Ovchinnikov. Similarity relations, fuzzy partitions, and fuzzy orderings. *Fuzzy Sets and Systems*, 40:107–126, 1991.

[10] G.M. Reaven and R.G. Miller. An inquiry into the nature of diabetes mellitus using a multidimensional analysis. Technical Report 33, Division of Biostatistics, Stanford University, 1979.

[11] Ronald R. Yager. Connectives and quantifiers in fuzzy logic. *Fuzzy Sets and Systems*, 40:39–75, 1991.

[12] Lotfi A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[13] Lotfi A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning. *Information Sciences*, 8:199–249, 1975.

[14] Lotfi A. Zadeh. Fuzzy sets and their application to pattern classification and clustering analysis. In J. van Ryzin, editor, *Classification and clustering*, pages 251–299. Academic Press, New York, 1978.

[15] H. J. Zimmermann and P. Zysno. Latent connectives in human decision making. *Fuzzy Sets and Systems*, 4:37–51, 1980.