Competitive and Hybrid Neuro-Fuzzy Models for Supervised Classification

Nicola Giusti Alessandro Sperduti
Dipartimento di Informatica, Università di Pisa,
Corso Italia 40, 56125 Pisa, ITALY
perso@di.unipi.it

Francesco Masulli
Dipartimento di Fisica, Università di Genova,
Via Dodecaneso 33, 16146 Genova, ITALY
masulli@ge.infm.it

Abstract

Neuro-fuzzy systems are often very complex and may require long training times. In the context of supervised classification, we propose a competitive and a hybrid model based on Fuzzy Basis Function networks. These models are fast to train and still hold very good generalization performances. Experimental results on the classification of handwritten digits are presented.

1 Introduction

The development of Fuzzy Logic Systems (FLS) requires the definition and tuning of the shapes and sizes of the membership functions. Neuro-fuzzy systems allow the automatic ajustment of these parameters on the basis of training data. Very often, however, the resulting systems are very complex and may require long training times. Thus, it is of paramount importance to devise simple systems which are fast to train and still hold very good generalization performances.

In this paper, we briefly review how FLS can be implemented by Fuzzy Basis Function networks (FBFN) and then, in the context of supervised classification, we show how to derive simple but effective models from the original formulation of a FBF network.

In Section 2 we discuss how FLS can be implemented by a neuro-fuzzy network. A simplified version of the model is discussed as well. New models, based on an extension of the simplified version and on the combination of the neuro-fuzzy network with the k-Nearest-Neighbour Rule [1] are introduced in Section 3. Experimental results obtained on the classification of handwritten digits are presented in Section 4. Conclusions are drawn in Section 5.

2 The Fuzzy Basis Function network

A Fuzzy Logic System with *singleton* fuzzification, *max-product* composition, *product inference* and *height* defuzzification can be represented as [7]

$$y = f(\mathbf{x}) = \sum_{l=1}^{M} \overline{y}^{\,l} \phi_l(\mathbf{x}) \qquad (1)$$

with

$$\phi_l(\mathbf{x}) = \frac{\prod_{i=1}^{p} \mu_{F_i^l}(x_i)}{\sum_{l=1}^{M} \prod_{i=1}^{p} \mu_{F_i^l}(x_i)} \qquad (2)$$

where $\overline{y}^{\,l}$ denote the center of gravity of the output fuzzy set, and $\phi_l(\mathbf{x})$, $l = 1, 2, ..., M$, are called *fuzzy basis functions*. We can refer to those FLS as *fuzzy basis expansions* or *networks of fuzzy basis functions* (FBF network)[1].

The relationships between fuzzy basis expansions and other basis functions have been extensively studied in [3]. It is worth noting that the FLS with universal function property studied by Mendel and Wang [9, 8] (i.e., a singleton FLS using product inference, product implication, Gaussian membership and height defuzzification) can be rewritten as a FBF network expansion.

In this paper we use as reference model a neuro-fuzzy logic system based on a multi-input-multi-output (MIMO) version of this FBF network. Specifically, if there are K units in the input layer, J fuzzy inference rules and I outputs, the rule activations can be expressed as:

$$r_j = \prod_k \mu_{jk}(x_k) \qquad (3)$$

$$\mu_{jk}(x_k) = \exp\left(-\frac{(x_k - m_{jk})^2}{2\sigma_{jk}^2}\right) \qquad (4)$$

$$y_i = \frac{\sum_j r_j \overline{y}_{ij}}{\sum_j r_j} = \sum_j \overline{y}_{ij}\phi_j(\mathbf{x}) \qquad (5)$$

$$\phi_j = \frac{\prod_k \mu_{jk}(x_k)}{\sum_j \prod_k \mu_{jk}(x_k)} \qquad (6)$$

where the quantity $\mu_{jk}(x_k)$ represents the value of the membership function of the component x_k of the input vector for

[1] In [7] fuzzy basis expansions for FLS with nonsingleton fuzzification are also introduced.

0-7803-4122-8/97 $10.00©1997 IEEE

the jth rule, m_{jk} and σ_{jk}^2 are the means and the variances of the Gaussian membership functions, y_i are the values of the output units, $\overline{y}_{ij}$ is the center of gravity of the output fuzzy membership function of the jth rule associated with the output y_i, and ϕ_j is the fuzzy basis function associated to rule j, representing its normalized activation [2].

The FBF network can be regarded as a feedforward connectionist system with one hidden layer whose units correspond to the fuzzy rules. The FBF network can be identified both by exploiting the linguistic knowledge available (*structure identification problem*)[5] and by using the information contained in a data set (*parameter estimation problem*) [5]. Learning rules based on Gradient Descent technique are discussed, e.g., in [9].

2.1 The Simplified FBF Network

For pattern recognition applications, from this FBF network a *Simplified FBF network* (SFBF network) can be obtained by assuming, in accordance with *rule specialization* [6]:

$$\overline{y}_{ij} \equiv \delta_{ij} = \begin{cases} 1 & \text{if rule } j \text{ is} \\ & \text{associated to class } i, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

This assumption leads to both a system with as many units as classes and a strong simplification of the learning formulas, that become:

$$\Delta m_{jk} = \eta_m \phi_j U_{ij} [x_k - m_{jk}]/\sigma_{jk}^2 \tag{8}$$

$$\Delta \sigma_{jk} = \eta_\sigma \phi_j U_{ij} [x_k - m_{jk}]^2/\sigma_{jk}^3 \tag{9}$$

with

$$U_{ij} = \begin{cases} (y_i - 1)^2 & \text{if } j = i \\ y_i^2 - y_i & \text{if } j \neq i \end{cases} \tag{10}$$

It is worth noting that, from Equation 7 and the form of the defuzzifier, $y \in (0, 1)$ follows, and consequently

$$U_{ij} = \begin{cases} \geq 0 & \text{if } j = i \\ \leq 0 & \text{if } j \neq i \end{cases} \tag{11}$$

holds.

Therefore, the learning rules of the SFBF network are competitive. During training, the means of the Gaussian membership functions of each rule move towards the patterns of the class associated to that rule, and escape from patterns belonging to other classes. At the same time, sigmas of Gaussian membership functions of each rule grow in order to increase the value of the membership function for patterns of the class associated to that rule, or shrink in order to reduce the value of the Gaussian membership function for patterns belonging to other classes.

Of course, since this system must have as many units as classification classes, it cannot be used for complex classification tasks. In the next section, we present a new version of the model which removes this limitation.

[2]Without loss of generality, we could assume that the fuzzy membership functions are singletons ($\overline{y}_{ij} \equiv s_{ij}$).

3 New Models: Extended SFBF Network and Hybrid System

In this section we introduce two new models. The first one, i.e., the *Extended SFBF Network* (ESFBF) allows the use of an arbitrary number of units in the SFBF Network, while the second one, i.e., the *Hybrid System* (HS), improves the generalization capability of the previous one by applying, in a non trivial way, the *Nearest Neighbour* rule on the rejected patterns.

3.1 Extended SFBF Network

To remove the constraint on the number of units in the SFBF network, we introduce a new level of competition among units. The new defined network possesses n_j units associated to each class j, for a total of $J = \sum_{j=1}^{I} n_j$ units. During learning the output of each unit is computed and the best unit for each class is selected, i.e., for each class j the unit $i_j^* \in Idx_j = \{1, \ldots, n_j\}$ such that $i_j^* = \underset{i \in Idx_j}{arg\,max}\{\phi_i\}$ is selected. In that way the number of selected units is equal to the number of classes and the learning rules of the SFBF network can be applied. Thus, at each learning step, only the selected rules have the weights changed. During the operational phase, the input pattern is classified by the class label associated with the unit having maximum activity.

3.2 Hybrid System

In order to improve the generalization capability of the ESFBF network, we studied a hybrid pattern recognition scheme based on a hierarchy made up by an ESFBF network with rejection, followed by a Nearest-Neighbor Rule (NR) classifier working on the patterns rejected by the ESFBF network. Specifically, after the training of the ESFBF network, a rejection rule is implemented, consisting in a *rejection threshold* on the level of the higher output, i.e., if no output of the ESFBF network is greater than the threshold, the pattern is rejected from the ESFBF network. The basic idea is that patterns rejected by the rejection rule with threshold value γ are then classified by the Nearest-Neighbor Rule with reference database made up by patterns rejected by the same ESFB, but using a threshold value $\lambda \geq \gamma$. By using the recognition threshold γ, the ESFBF network classifies very quickly most of the patterns with small classification

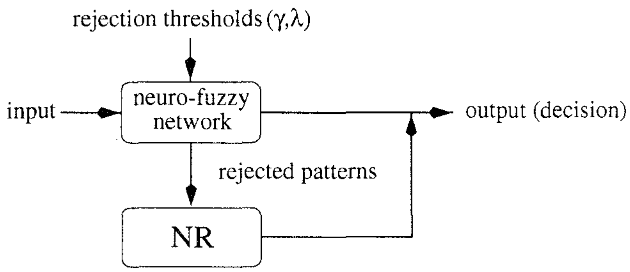


Figure 1. Hybrid Pattern Recognition Scheme combining the SFBF network and the NR.

MODEL	%S-Valid.	Epochs	Ep. Dur. (sec)
NR_1	92.89	–	–
$FBFN_{48}$	94.09	13	1925
$FBFN_{12}$	92.26	36	307
$FBFN_{10}$	92.23	55	180
SFBFN	91.36	10	30

Table 1. Comparison among NR with $k = 1$, FBF networks (with 48, 12, and 10 units), and SFBF network. %S-Validation is the success rate on the validation set. The epoch duration is measured on a Sun 10.

error, while a minority of patterns are forwarded to the NR for classification. Moreover, the quality of the NR database is controlled independently by the threshold λ. Of course, for rejected patterns, the recognition speed depends mainly on the dimension of the NR database. However, to speed-up the recognition time of the NR classifier, at classification time one can use an efficient *on-line editing strategy*, by considering for the NR only patterns belonging to classes that get the first L higher rates by the ESFBF network.

4 Experimental Results

We tested the performance of the above systems on the classification of handwritten digits. We used a training set, a test set, and a validation set extracted from the NIST-3 database [2]. Both the training set and the test set was made up of 10,000 associative pairs of segmented handwritten digits each, obtained from disjoint groups of writers, while the validation set consisted of 5,000 independent digits.

The preprocessing included the following steps: 1) digit image extraction from the CD-ROM and normalization to a 32×32 binary matrix; 2) low-pass filtering in order to remove some small spots and holes from the image; 3) application of a shear transform to the digit image to straighten the axis joining the first upper-left point of the digit image to the last lower-right point; 4) image skeletonization by using a thinning algorithm; 5) finally, transformation of the digit representation into a 64-element vector, each vector element representing the number of black pixels contained in adjacent 4×4 squares (local counting). It is worth noting that the resulting digit representation exhibits sufficient degrees of invariance to both scale and small image shifts or rotations.

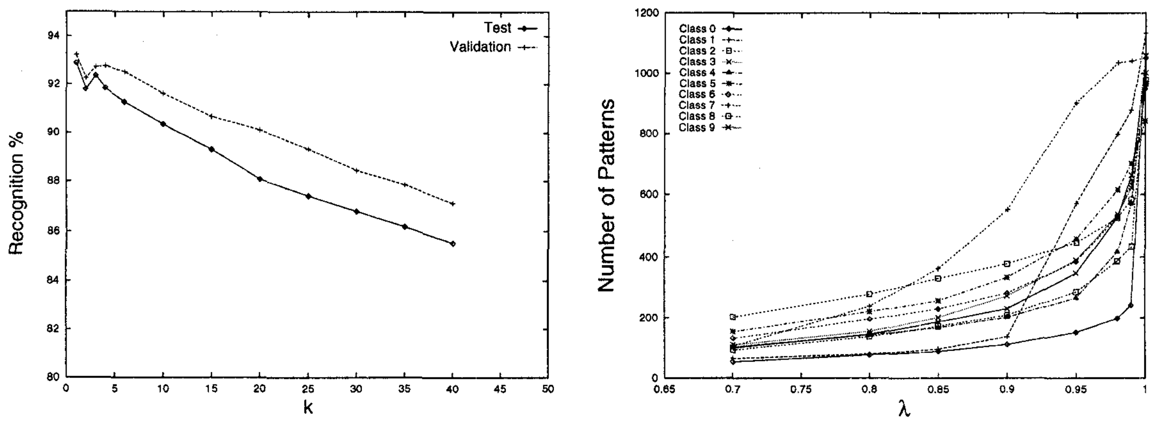
For comparison purposes, in Table 1 we have reported the performance of NR, FBF networks with different number of units, and SFBF network on the validation set. The training of the networks was stopped using the test set (early stopping). Note that, while the SFBF network is not able to reach the same generalization performaces of the FBF networks, its training is much faster.

In Figure 2-a we have reported the performance of the k-Nearest-Neighbour for different values of k. Due to noise in the data, the best performance was obtained for $k = 1$. Moreover, in Figure 2-b, we have reported the distribution for classes of the reference database for the NR ($k = 1$) (to be used in the HS) generated by a ESFBF network with 20 units (2 for each class) for different values of λ. The training and test curves for this network are reported in Figure 3-a, while the effect of the rejection threshold on the training and test set for the same ESFBF network is shown in Figure 3-b. The error of the network on the validation set was 6.2%. It must be noted that the performance of this network was very close to the performance of the FBF network with 48 units, thus showing that the ESFBF networks, besides to be much faster in training, can reach generalization performances which are comparable to the ones obtained by more complex models.

The same ESFBF network has then been used in a HS with $k = 1$ for the NR. In Figure 4 we have reported the test and validation curves for $\gamma = 0.96$ and different values of λ. As expected the performances of the system improved with the dimension of the reference database for the NR (higher values of λ). On the contrary, the performances did not depend linearly on the number of classes used for the NR (on-line editing). To assess the "best" values for γ and L (i.e., the number of classes to be used by the NR) we have performed experiments with $\lambda = 1.0$ and different values for γ and L. The results obtained for the test and validation sets are reported in Figure 5. The test set got the best ratings with $\gamma = 0.88$ and $L = 7$, while the validation set reached the best performances with $\gamma = 0.88$ and $L = 2$.

5 Conclusion

In the context of supervised classification we have presented two new neuro-fuzzy models. The first model is



a) b)

Figure 2. a) Performance of the k-Nearest-Neighbour on the test and validation sets using as reference database the training set; b) Composition of the reference database for the NR ($k = 1$) for different classes and λ values. Note that the database for $\lambda = 1.0$ is equal to the training set.

an extension of the Simplified FBF Network which allows the use of an arbitrary number of units. The second one is an hybrid system which combines the previous model with the Nearest-Neighbour Rule. The main advantage of these models is that they are simple and fast to train. Moreover, they hold good generalization performances. Specifically, the hybrid system improves on the performances of the former one at the expenses of a slower response time. This result is in agreement with the results obtained by related hybrid systems (see for example [4]).

References

[1] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* New York: J. Wiley & Sons, 1973.

[2] M. Garris and R. Wilkinson. Nist special database3 handwritten segmented characters. Technical report, National Institute of Standard and Technology, Gaithesburg, MD , USA, 1992.

[3] H. Kim and J. Mendel. Fuzzy basis functions: Comparisons with other basis functions. *IEEE Transactions on Fuzzy Systems*, 3:158–168, 1995.

[4] S. Knerr and A. Sperduti. Rejection driven hierarchy of neural network classifiers. In *International Symposium on Nonlinear Theory and its Applications '93*, pages 957–961, 1993. Waikiki, USA.

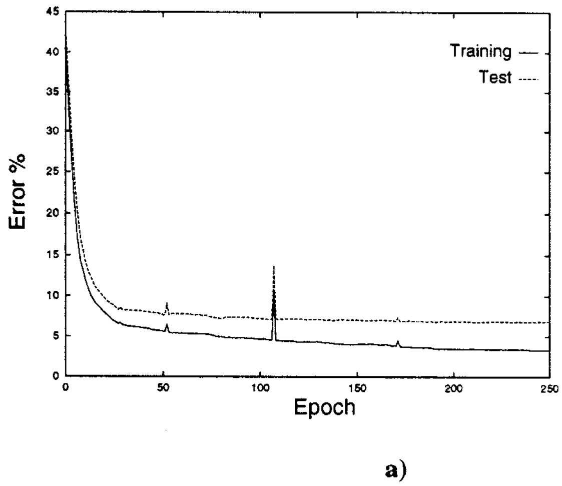
[5] C. Lee. Fuzzy logic in control systems: fuzzy logic controller. I. *IEEE Transactions on Systems, Man and Cybernetics*, 20:404–418, 1990.

[6] F. Masulli and A. Sperduti. Competitive learning in a classifier based on an adaptive fuzzy system. In P. Anderson and K. Wawick, editors, *Proceedings of the International ICSC Symposium on Industrial Intelligent Automation (IIA'96) and Soft Computing (SOCO'96)*, pages C2–C8, 1996.

[7] J. Mendel. Fuzzy logic systems for engineering: A tutorial. *Proceedings of the IEEE*, 83:345–377, 1995.

[8] L. Wang and J. Mendel. Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *IEEE Transaction on Neural Networks*, 5:807–814, 1992.

[9] L. Wang and J. Mendel. Generating fuzzy rules by learning from examples. *IEEE Transaction on Systems, Man, and Cybernetics*, 22:1414–1427, 1992.



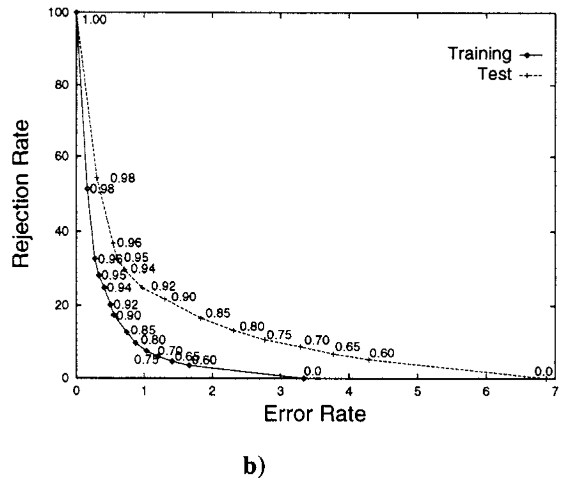


Figure 3. a) **Training and test curves for a ESFBF with 20 units (2 units for each class);** b) **Effect of the rejection threshold (reported near each experimental point) on the training and test set for the same ESFBF.**

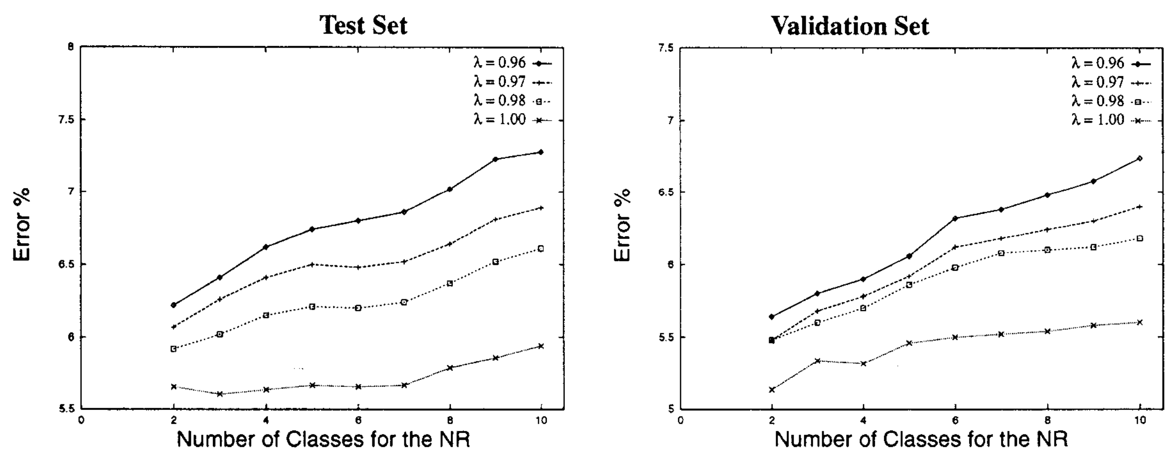


Figure 4. Error curves for the Hybrid System (ESFBF with 20 units) with $\gamma = 0.96$ and different values of λ.

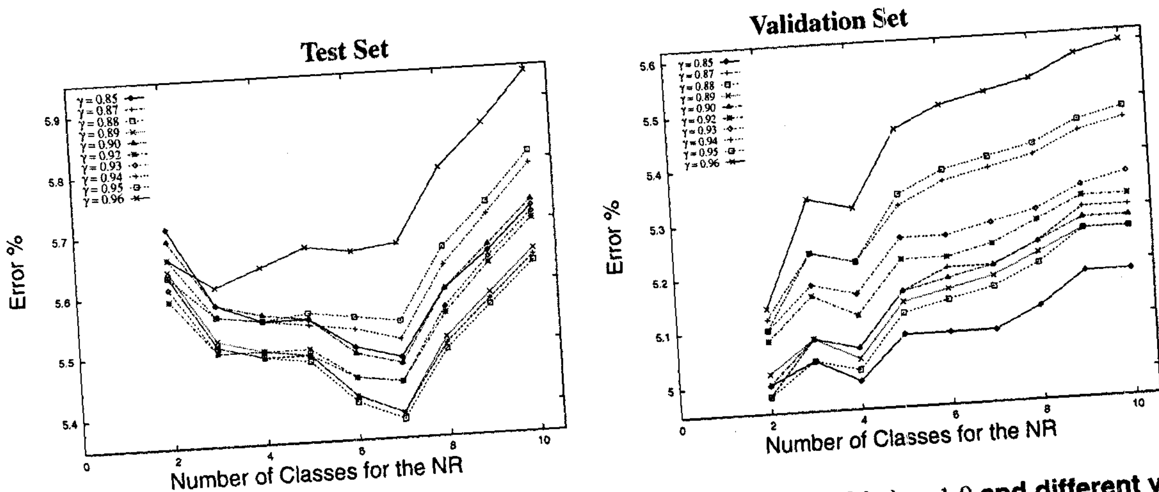


Figure 5. Error curves for the Hybrid System (ESFBF with 20 units) with $\lambda = 1.0$ and different values of γ.