PROCEEDINGS REPRINT

SPIE—The International Society for Optical Engineering

Reprinted from

Applications of Fuzzy Logic Technology III

10–12 April 1996
Orlando, Florida

SPIE
P
PROCEEDINGS SERIES

Volume 2761

Invited Paper

A Hybrid Pattern Recognition Scheme

Francesco Masulli

Istituto Nazionale di Fisica della Materia
Dipartimento di Fisica - Università di Genova
Via Dodecaneso 33 - 16146 Genova - Italy
email : masulli@ge.infm.it

Alessandro Sperduti

Department of Computer Science - University of Pisa
Corso Italia, 40 - 56125 Pisa - Italy
email : perso@di.unipi.it

Davide Alfonso

Dipartimento di Informatica e Scienze dell'Informazione
Università di Genova
Via Dodecaneso 35 - 16146 Genova - Italy
email : alfonso@disi.unige.it

Abstract

Networks of *Fuzzy Basis Functions* (FBF) characterized by singleton fuzzifier, Gaussian membership functions, product-inference, and height method defuzzifier, show interesting characteristics, including the approximation of the Bayes discriminant function. In this paper, a classifier based on a *Simplified FBF (SFBF) network* is presented and its performances are studied in the framework of handwritten digits recognition. The learning rules of the SFBF network are less complex than those of a FBF network, and experimental results show a significant speed-up of learning, at the cost of a small decrease of the generalization performances. Moreover, a ***hybrid pattern recognition scheme*** (HS) is proposed, based on a hierarchy of a SFBF network plus a Nearest-Neighbor Rule (NR), that recognizes the patterns rejected by the SFBF network. This approach permits to recover the loss in generalization exhibited by the SFBF network alone. Specifically, the efficiency of the hierarchy can be improved, since the output of the SFBF network for a rejected test pattern can be used to edit the set of rejected (training set) patterns: the NR searches only for patterns belonging to classes that get the highest rates by the SFBF network.

Keywords: Fuzzy Basis Function Networks; Classification; Hybrid Models; k-Nearest Neighbor Rule; Handwritten Character Recognition.

0-8194-2142-1/96/$6.00

1 INTRODUCTION

In previous papers[10,8,9] we studied the behavior of an adaptive fuzzy logic system realized as a network of *Fuzzy Basis Functions* (FBF), and characterized by singleton fuzzifier, Gaussian membership functions, product-inference, and height method defuzzifier. This fuzzy logic system holds the universal function approximation property and the capability of learning from examples[6,14,10,8] (details are presented in another paper of this book[7]). Moreover a non-parametric classifier able to approximate the Bayes discriminant function can be based on it. It is worth noting that a classifier based on a fuzzy logic system can exploit the information given as a numerical training set and/or as a linguistic description of classes.

Experimental results show that, when only numerical data is available, the choice of a proper structure for an FBF network (structure identification problem) is bounded on the number of fuzzy rules.[2,1] Specifically, we have shown that for a classification problem, the number of fuzzy rules of the FBF network has a lower bound which corresponds to the number of classes. Indeed, an FBF network with less rules than classes is unable to obtain a reasonable classification performance because of the inability of the system to recognize whole classes, i.e., each rule specializes itself on a specific class. When the number of rules is increased up to the number of classes, a sharp increase of the performance is observed (*semantic phase transition*). Thus, even if the FBF network is organized as a supervised feedforward network, its behavior is closer to the one of a *competitive model* showing a strong specialization of the fuzzy rules. A pruning technique was proposed by Casalino et al.,[2] in order to automatically remove less relevant fuzzy rules from an oversized system.

In this paper, a classifier based on a *Simplified FBF (SFBF) network*, showing a competitive learning behavior, is presented and its performances are studied in the framework of handwritten digits recognition. The learning rules of the SFBF network are less complex than those of the FBF network, and experimental results show a significant speed-up of learning, at the cost of a small decrease of the generalization performances.

Moreover, a hybrid pattern recognition scheme (HS) is proposed, based on a hierarchy of a SFBF network plus a Nearest-Neighbor Rule (NR),[4] that recognizes the patterns rejected by the SFBF network. This approach permits to recover the loss in generalization exhibited by the SFBF network alone. Specifically, the efficiency of the hierarchy can be improved, since the output of the SFBF network for a rejected test pattern can be used to edit the set of rejected (training set) patterns: the NR searches only for patterns belonging to classes that get the highest rates by the SFBF network.

In the next section, the Simplified FBF network is presented. In Section 3, the data sets and the preprocessing of patterns are described. An experimental comparison between FBF network and SFBF network is reported in Section 4. In Section 5, we define the hierarchy and report its performances on the experimental data. Discussion and conclusions are presented in Section 6.

2 THE SIMPLIFIED FBF NETWORK

We consider a multi-input-multi-output (MIMO) fuzzy logic system based on network of Fuzzy Basis Functions (FBF) using singleton fuzzifier, Gaussian membership functions, product-inference, and height method defuzzifier.[7] We shall denote the input and the output values of the FBF network as x_k and y_i, respectively,the means and the variances of the Gaussian membership functions as m_{jk} and σ^2_{jk}, and the maximum value of the output fuzzy membership functions (that, without loss of generality, can be chosen as singletons) as s_{ij} (with $k \in [1, K]$, $j \in [1, J]$, $i \in [1, I]$).

For pattern recognition applications, from this FBF network a *Simplified FBF network* (SFBF network) can be obtained by assuming, in accordance with *rule specialization*[2]:

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

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

$$\Delta m_{jk} = \eta_m \psi_j \Upsilon_{ij} [x_k - m_{jk}]/\sigma_{jk}^2 \tag{2}$$

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

with

$$\Upsilon_{ij} = \begin{cases} (y_i - 1)^2 & \text{if } j = i \\ y_i^2 - y_i & \text{if } j \neq i \end{cases} \tag{4}$$

and ψ_j are *fuzzy basis functions*[13]

$$\psi_j = \frac{\prod_k \mu_{jk}(x_k)}{\sum_j \prod_k \mu_{jk}(x_k)}. \tag{5}$$

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

$$\Upsilon_{ij} = \begin{cases} \geq 0 & \text{if } j = i \\ \leq 0 & \text{if } j \neq i \end{cases} \tag{6}$$

holds.

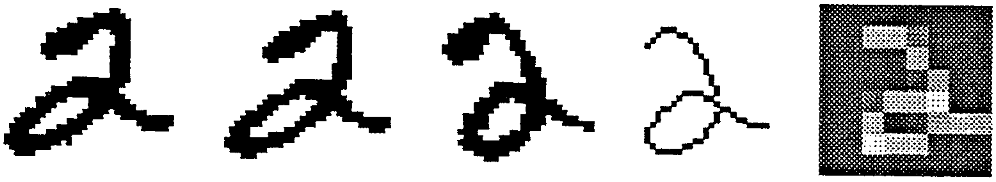
Therefore, the learning rules of the SFBF network are competitive. During training, the means of the Gaussian membership functions of each rule move towards the patterns of the class associated to that rule, and escape from patterns belonging to other classes. At the same time, sigmas of Gaussian membership functions of each rule grow in order to increase the value of the membership function μ 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. An interesting property of these new learning formulas is that they reduce of one order of magnitude the complexity of training.

3 DATA BASE AND PREPROCESSING

All the experiments reported in the following sections were carred out on a SUN 10/50 workstation. We used a training set, a test set, and a validation set extracted from the NIST-3 data-base.[5] The NIST-3 data-base, distributed on a CD-ROM, contains 313389 characters coded as 128×128 binary-matrix images and labeled by the corresponding ASCII codes.

Each of the training set, the test set and the validation set was made up of 10,000 associative pairs of segmented handwritten digits each, obtained from disjoint groups of writers.

As shown in Figure 1, the preprocessing included the following steps:



(a) (b) (c) (d) (e)

Figure 1: Preprocessing steps for a handwritten digit: Normalization (a), low-pass filtering (b), shear transform (c), skeletonization (d), local counting (e).

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[12];
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.

4 PERFORMANCES OF THE SFBF NETWORK

In the first experiment, we compared the performances of a SFBF network with those of FBF networks containing different numbers of rules and those of the Nearest-Neighbor Rule (NR).[4]

Before learning, for each FBF network, the values of each s_{ij}, m_{jk}, σ_{jk} were initializated at random. For the SFBF network only the s_{ij} were initializated at random, while σ_{jk} were initializated at the value .25, and m_{jk} as

$$m_{jk} = x_k^n \ , \tag{7}$$

where $\mathbf{x^n} = \{x_k^n\}$ is an example of class j randomly extracted from the training set.

MODEL	%S-Validation	Epochs	Epoch Duration (sec)
NR	94.07	–	–
FBF_{48}	94.09	13	1925
FBF_{12}	92.26	36	307
FBF_{10}	92.23	55	180
SFBF	91.36	10	30

Table 1: Comparison among the NR, the FBF network (with 48, 12, and 10 rules), and the SFBF network. %S-Validation is the success rate on the validation set.

For the FBF networks and the SFBF network, the respective gradient descent algorithms were applied by following a training by pattern method. To avoid overfitting, the training was stopped when the errors on the test set reached the minimum value. In Table 1, for each pattern recognition model, the success rate on the validation set (%S-Validation), the number of training epochs (Epochs), and the duration of one training epoch (Duration/Epochs), are listed. Even if the success rates on the NR and the FBF_{48} were equivalent, the mean recognition time per pattern of the NR was .733 sec, while for the FBF_{48} it was .004 sec. Moreover, the generalization performances of the FBF_{10} network and the SFBF network were similar, while the SFBF network resulted to be faster in learning, in accordance with the lower complexity of the learning rules.

5 HYBRID SCHEME

In order to recover the loss in generalization exhibited by the SFBF network, we studied a hybrid pattern recognition scheme (HS) based on a hierarchy made up by an SFBF network with rejection, followed by a Nearest-Neighbor Rule classifier working on the patterns rejected by the SFBF network.

After the training of the SFBF network, a rejection rule is implemented, consisting in a *rejection threshold* on the level of the higher output. If no output of the SFBF network is greater than the threshold, the pattern is rejected from the SFBF network and the Nearest-Neighbor Rule classifier is applied to it. By using the recognition threshold, the SFBF network classifies very quickly most of the patterns with small classification error, while a minority of patterns are forwarded to the NR for classification. For rejected patterns, the recognition speed depends mainly on the dimension of the space of search used by the NR.

In order to speed-up the recognition time of the NR classifier, we studied some optimizations of the hybrid scheme, consisting in editing strategies able to reduce the dimensions of the data-base to be used by the NR.

The first optimization consisted in an *off-line editing strategy* to be used at the completion of the learning of the SFBF network. This method consists in a condensation of the training set, obtained by filtering the original training set by the SFBF network itself, with the application of the rejection threshold to the classification algorithm. By using this strategy, for each value of the rejection threshold on the SFBF network output, one obtains a *condensed data base* (CDB) containing only the patterns close to the decision surfaces, that are the most important for classification. In Figure 3, for the SFBF network, the rejection rate of patterns of the training set, versus the error rate on the accepted (classified) patterns, is plotted. Near each experimental point, the value of the corresponding rejection threshold applied to the SFBF network, is reported. This information is used for the selection of the most suitable threshold to be used within the hybrid scheme.

As shown in Table 2, the value of the rejection threshold (Rej-Threshold) affects the overall performances of the hybrid scheme. In this table CDB-Dimension stands for the dimension of the resulting Condensed Data base, %E-Training is the error rate on the accepted patterns of the training set, %S-Validation is the success rate on the validation set, and Rec-Time is the mean recognition time for pattern.

According to Tables 1 and 2, the hybrid scheme reaches generalization performances similar to those of the NR (using the full training set of 10,000 examples), and of the FBF_{48}. Moreover, for the hybrid scheme, the learning time is the same as for the SFBF network, and, therefore, is smaller than for the FBF_{48} (and for the FBF_{10}), while recognition times are smaller than that of the NR alone . From Table 2, one could conclude that, improvements of the recognition times with the hybrid scheme can be obtained to the detriment of the generalization rate, or vice-versa. Whereas, at classification time one can use an efficient *on-line editing strategy*, by considering for the Nearest-Neighbor Rule (NR) only patterns belonging to classes that get the first L higher rates by the SFBF network.

In Table 3 for each value of L ($L \in [2, 10]$), the success rate on the validation set (%S-Validation) is shown.

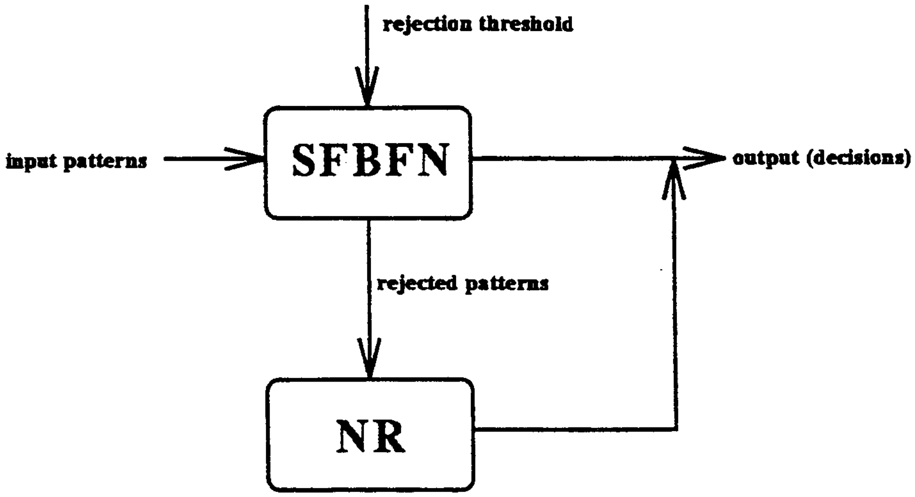
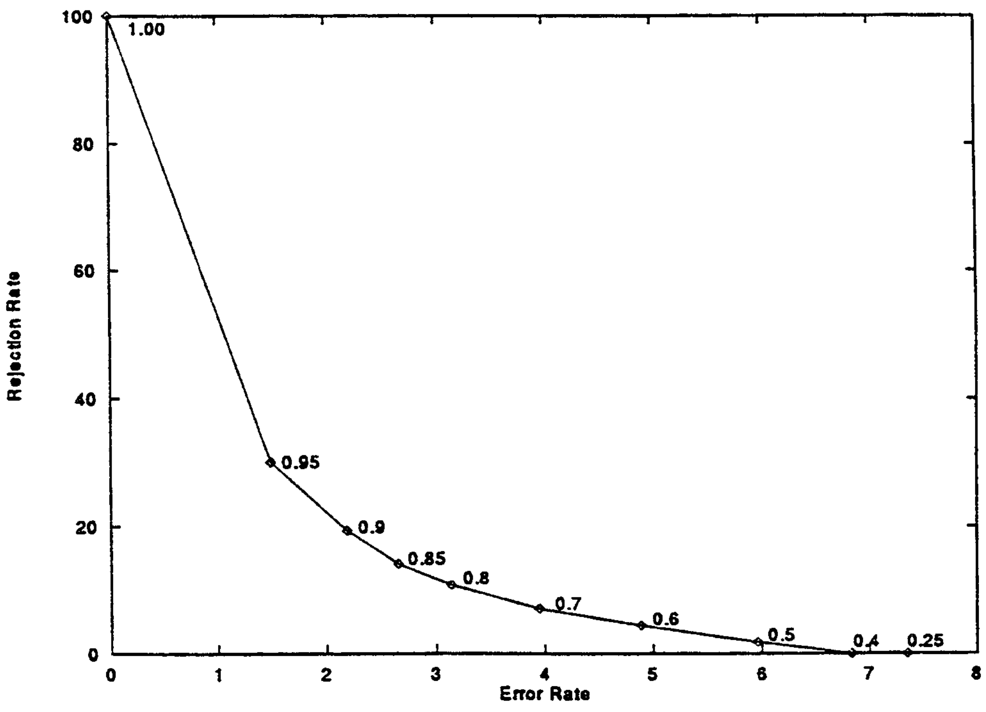


Figure 2: Hybrid Pattern Recognition Scheme combining the SFBF network and the NR.



gure 3: Effect of the rejections threshold (reported near each experimental point) on the training set.

Rej-Threshold	CDB-Dimension	%E-Training	%S-Validation	Rec-Time (sec)
.98	4549	.89	93.62	.0750
.95	3019	1.49	93.45	.0324
.70	713	3.95	92.43	.0020

Table 2: Hybrid Pattern Recognition Scheme: Performances with different rejection thresholds (see text).

L	%S-Validation
2	93.46
3	93.53
4	93.68
5	93.58
6	93.64
7	93.66
8	93.61
9	93.60
10	93.45

Table 3: Hybrid Pattern Recognition Scheme: Performances with on-line edited Condensed Data Base (see text).

The rejection threshold for each experiment is set to .95. These experimental data point out that there is no significant dependence of the success rate on L. As a consequence a choice of $L = 2$, leads to good generalization performances and the higher recognition speed (about 5 time faster than for L=10, with our preliminary results).

6 DISCUSSION AND CONCLUSION

In this paper, a classifier based on a *Simplified FBF network*, showing a competitive learning behavior, is presented and its performances are studied in the framework of handwritten digits recognition. The learning rules of the SFBF network are less complex than those of the FBF network, and experimental results show a significant speed-up of learning, at the cost of a small decrease of the generalization performances.

Moreover, a hybrid pattern recognition scheme (HS) is proposed, based on a hierarchy of a SFBF network plus a Nearest-Neighbor Rule (NR), that recognizes the patterns rejected by the SFBF network. This approach permits to recover the loss in generalization exhibited by the SFBF network alone. Specifically, the efficiency of the hierarchy can be improved, since the output of the SFBF network for a rejected test pattern can be used to edit the set of rejected (training set) patterns: the NR searches only for patterns belonging to classes that get the highest rates by the SFBF network.

In order to obtain a further speed-up in the search for the closest points by the NR, a parallel hardware can be developed, and/or the usage of optimized data structures, such as Bumptrees,[11] or randomized algorithms,[3] can be experimented.

Among many possible developments of the proposed hybrid pattern recognition scheme, an incremental version of this system can be devised without any relevant changes to the system. The idea is that as soon as an additional training example is available, it is classified by the SFBF network. If the example is correctly classified and with a good confidence, i.e., at least one output is above the threshold, then the system do not need to be updated. Otherwise, the example is rejected by the SFBF network and inserted into the database of the NR classifier. This basic version can be extended to controlling the increase in the NR's database as follows. When the size of the

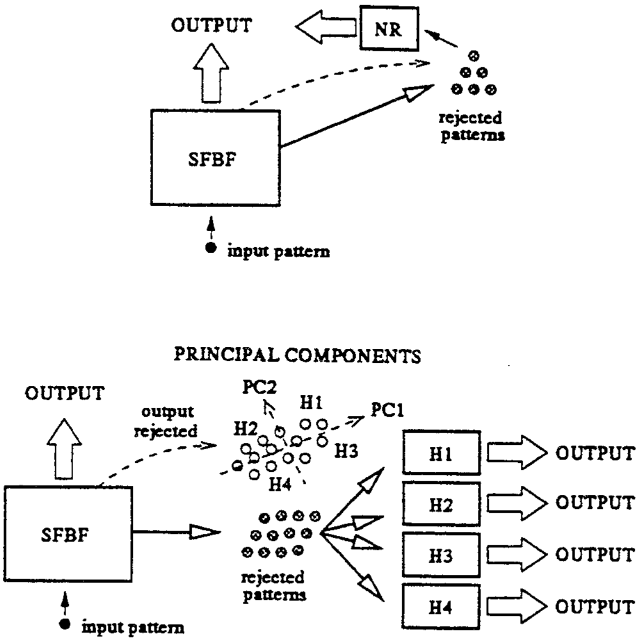


Figure 4: **Top:** Hybrid System. **Bottom:** Incremental Hybrid System.

NR's database becomes too large, it is split in the following way: all the patterns in the database are presented to the SFBF network and the corresponding outputs of the SFBF network are collected into an output database ODB. Then, the first two principal components, PC_1 and PC_2, for ODB are computed and the ODB split in four subsets, ODB_1, ODB_2, ODB_3, ODB_4, according to the sign of the projections of each pattern in ODB with respect to PC_1 and PC_2. Each subset ODB_i corresponds to a subset DB_i of the patterns in the NR's database. This subset is then used as original training set for a new hybrid system H_i. During the operational phase, if a pattern p_x is rejected by the original SFBF network, then its output $o(p_x)$ is projected onto the principal components PC_1 and PC_2 and the result used to route p_x towards the most appropriate hybrid system H_i for classification.

The reason for this construction is that the NR's database is usually characterized by strong localities which makes it very difficult to learn by a new SFBF network for further classification. By using the output of the original SFBF network for splitting it, we are able to select subsets of patterns which are locally related. Principal components are used in order to reduce the potential error induced by the splitting. Moreover, it should be possible to reduce this potential error by allowing the subsets to overlap. This can be realized by having a threshold on the modulus of the projection on PC_1 and PC_2. In this way, some boundary patterns may belong to more than one subset.

7 ACKNOWLEDGMENTS

This work was supported by grants from CNR-Progetto Strategico Reti Neurali, GNCB-CNR, INFM, and MURST.

8 REFERENCES

[1] F. Casalino, F. Masulli, and A. Sperduti. Rule specialization and semantic phase transition in the adaptive fuzzy system. In N.C. Steele, editor, *Proceedings of the ICSC International Symposium on Fuzzy Logic*, pages B87–B92, Zurich (Switzerland), 1995. ICSC Academic Press - Millet-Alberta, Canada.

[2] F. Casalino, F. Masulli, A. Sperduti, and F. Vannucci. Semantic phase transition in a classifier based on an adaptive fuzzy system. In *Proceedings of the Third IEEE International Conference on Fuzzy Systems, IEEE-FUZZ94*, volume 2, pages 808–812, Orlando, FL, USA, 1994. IEEE.

[3] K. L. Clarkson. A randomized algorithm for the closest-point queries. *SIAM Journal on Computing*, pages 830–847, 1988.

[4] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* Wiley, New York, 1973.

[5] M.D. Garris and R.A. Wilkinson. *NIST Special Database3 Handwritten Segmented Characters.* National Institute of Standard and Technology, Gaithesburg, MD , USA, 1992.

[6] C.C. Jou. Comparing learning performance of neural networks and fuzzy systems. In *IEEE International Conference on Fuzzy Systems*, pages 1028–1033, San Francisco, 1993. IEEE, New York, NY.

[7] Castellano. M, F. Masulli, and M. Penna. Fuzzy systems in high energy physics. In B. Bosacchi and J. Bezdek, editors, *Applications Applications of Fuzzy Logic Technology III - SPIE Proceedings Series*, Orlando, Florida, 1996. SPIE - Bellingam, WA, USA.

[8] F. Masulli. Bayesian classification by feedforward connectionist systems. In F. Masulli, P. G. Morasso, and A. Schenone, editors, *Neural Networks in Biomedicine - Proceedings of the Advanced School of the Italian Biomedical Physics Association - Como, Italy , 1993*, pages 145–162, Singapore, 1994. World Scientific.

[9] F. Masulli and F. Casalino. A neuro-fuzzy system for Bayesian classification. In *Proceedings of EUFIT'95 Third European Congress on Intelligent Techniques and Soft Computing*, pages 1425–1429, Aachen (Germany), 1995. Verlag und Druck Mainz.

[10] F. Masulli, F. Casalino, and F. Vannucci. Bayesian properties and performances of adaptive fuzzy systems in pattern recognition problems. In M. Marinaro and P.G. Morasso, editors, *Proceedings of the European Conference on Artificial Neural Networks, ICANN-94*, pages 189–192, Sorrento, Italy, 1994. Springer.

[11] S. M. Omohundro. Bumptrees for efficient function, constraint, and classification learning. In R.P. Lippmann, J.E. Moody, and D. S. Touretzky, editors, *Advances in Neural Information Processing Systems 3*, pages 693–900. San Mateo, CA: Morgan Kaufmann, 1991.

[12] Theo Pavlidis. *Algorithms for Graphics and Image Processing.* Springer-Verlag, 1982.

[13] L. Wang and J.M. Mendel. Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *IEEE Trans. on Neural Networks*, 5:807–14, 1992.

[14] L. X. Wang. *Adaptive Fuzzy Systems and Control.* Prentice Hall, Englewood Cliffs, New Jersey, 1994.