Rule Specialization in Networks of Fuzzy Basis Functions

F. Casalino[(1)] F. Masulli[(1)] A. Sperduti[(2)]

(1) Istituto Nazionale per la Fisica della Materia and
Dipartimento di Fisica - Università di Genova
Via Dodecaneso 33, 16146 Genova, ITALY
E-mail: {masulli|casal}@ge.infm.it

(2) Dipartimento di Informatica - Università di Pisa
Corso Italia 40, 56125 Pisa, ITALY
E-mail: perso@di.unipi.it

Abstract

The structure identification of adaptive fuzzy logic systems, realized as networks of Fuzzy Basis Functions (FBF's) and trained on numerical data, is studied for a handwritten character recognition problem. An FBF network with fewer rules than classes to be discriminated is unable to recognize some classes, while, when the number of rules is increased up to the number of classes to be discriminated, a sharp increase in the performance is observed. Experimental results point out that the behavior of the FBF network is closer to that of a competitive model showing a strong specialization of the fuzzy rules.

Key Words: fuzzy basis functions, neuro-fuzzy systems, structure identification, handwritten character recognition, rule specialization, semantic phase transition.

1 Introduction

A critical aspect of the design of a fuzzy inference system is the representation of knowledge as a set of fuzzy rules. Fuzzy rules can be obtained from linguistic descriptions by human experts or from the analysis of a physical model. However, the parameters of these linguistic descriptions are very often unknown, or even linguistic descriptions cannot be obtained and only a large amount of raw data are available. Thus, it is very important to device automatic techniques for fuzzy rule extraction from examples and numerical data. In the neural-network literature, several learning algorithms able to extract regularities from a data set have been

developed. In recent years, classes of fuzzy systems embedding some of these algorithms have been proposed for the purpose of deriving fuzzy rules from a data set (neuro-fuzzy systems). Some of these systems share important characteristics with neural networks, such as the Multi-Layer Perceptron (MLP) [22]; e.g., the feed-forward architecture, the capability of learning free parameters from numerical data sets, the universal function approximation capability [12, 8, 24, 25], and the approximation of the Bayes classifier [16]. Moreover, the available linguistic knowledge (even not complete) can be incorporated into a fuzzy inference system before the learning procedure, in order to speed-up the training phase.

In this paper, we study the structure identification problem in a Multi-Input-Multi-Output (MIMO) neuro-fuzzy system constituted by a network of *Fuzzy Basis Functions* (FBF's), previously presented in [8, 23, 9, 25], holding the universal function approximation property and the capability of learning from examples. In addition, the FBF network permits one to build a non-parametric classifier able to approximate the Bayes discriminant function. At a glance, FBF networks seem to be special cases of Radial Basis Function (RBF) networks or simple Gaussian expansions. This, however, has been shown not to be true by Kim and Mendel in [11], where they pointed out the differences between these models. Moreover, in [15], it has been shown experimentally that an FBF network can overfit the data, just like an MLP.

However, in spite of that and of the topological similarity between an FBF network and an MLP, the behavior of an FBF network is, in general, different from the one of an MLP. In particular, in a handwritten classification task presented in this paper, we observe that there is a minimal number of rules below which the system cannot perform the requested task and above which the system can perform the task very soon with a good generalization performance (*semantic phase transition* point). Specifically, in our classification problem, the generalization trend of the FBF network as a function of the amount of resources (hidden units) shows a sharp increase when passing from a system with 9 hidden units to a system with 10 hidden rules, i.e., with as many hidden units as the number of classes [2]. This behavior is in part justified by the locality of the activation functions used by the FBF network.

In the next section, the FBF network is presented. Then we discuss the structure identification problem (Section 3) and the semantic phase transition phenomenon (Section 5) in a classification problem (Section 4), as well as a comparison with MLP's trained on the same data set (Section 6). Conclusions are drawn in Section 7.

2 The FBF Network

We study a FBF network [8, 23, 9, 25] based on the following assumptions: height method defuzzifier, product-inference rule, singleton fuzzifier, and Gaussian membership function. Specifically, if there are K units in the input layer, J fuzzy inference rules and I outputs, the rule activations can be written as:

$$r_j \;=\; \prod_i \mu_{jk}(x_k), \tag{1}$$

The quantity $\mu_{jk}(x_k)$ represents the value of the membership function of the component x_i of the input vector for the j-th rule, and is defined as:

$$\mu_{jk}(x_k) = \exp\left(-\frac{(x_k - m_{jk})^2}{2\sigma_{ji}^2}\right), \tag{2}$$

where m_{jk} and σ_{ji}^2 are the means and the variances. The values of the output units are:

$$y_i = \frac{\sum_j r_j s_{ij}}{\sum_j r_j} \;, \tag{3}$$

and s_{ij} is the maximum value of the output fuzzy membership function of the $j-th$ rule associated with the output y_i. Without loss of generality, we assume that the fuzzy membership functions are singletons.

The FBF network can be organized as a feedforward connectionist system with just one hidden layer whose units correspond to the fuzzy MIMO rules.

In [8, 25], on the basis of the Stone-Weierstrass Theorem [21] the Universal Approximation Theorem was demonstrated that guarantees that an FBF network can perform approximation of continuous function at any assigned precision. As is well known, similar results on function approximation have been obtained by other feedforward connectionist systems, such as MLP's and RBF networks [4, 19].

The FBF network can be identified both by exploiting the linguistic knowledge available (*structure identification problem*) and by using the information contained in a data set (*parameter estimation problem*) [13].

For the FBF network, the parameter estimation problem can be solved by minimizing a suitable cost function, like the *mean square error* (MSE):

$$MSE = \frac{\sum_{i,n}(y_i^n - t_i^n)^2}{N} \;, \tag{4}$$

where N is the size of the training set, $\mathbf{y}^n = (y_i^n)$ is the network output, and $\mathbf{t}^n = (t_i^n)$ is the n-th label of the associative pair of the training set.

The cost function (4) can be minimized by many different techniques, among which the gradient descent technique, clustering methods [25], Kalman filters [7], genetic algorithms [3], etc. In our experiments, the FBF network parameters (i.e., m_{jk}, σ_{jk} and s_{ij}) were obtained by performing a gradient descent with respect to the MSE across the training set. The learning formulas are as follows [9, 25]:

$$\Delta s_{ij} = \eta_s[t_i - y_i]\psi_j \tag{5}$$

$$\Delta m_{jk} = \eta_m \psi_j \Upsilon_{ij} [x_k - m_{jk}]/\sigma_{jk}^2 \tag{6}$$

$$\Delta \sigma_{jk} = \eta_\sigma \psi_j \Upsilon_{ij} [x_k - m_{jk}]^2/\sigma_{jk}^3 \tag{7}$$

where

$$\Upsilon_{ij} = \sum_i [t_i - y_i][s_{ij} - y_i], \tag{8}$$

and the *fuzzy basis functions* [24]

$$\psi_j = \frac{\prod_k \mu_{jk}(x_k)}{\sum_j \prod_k \mu_{jk}(x_k)} \tag{9}$$

correspond to the normalized activations of the rules j, while η_s, η_m, and η_σ are the learning rates of s_{ij}, m_{jk}, and σ_{jk}, respectively.

Within this learning framework, a fuzzy non parametric classifier can be realized by training an FBF network with $\mathbf{t}^n$ defined as follows:

$$\mathbf{t}_i^n = \begin{cases} 1 & \text{if the pattern belongs to the class } i, \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

The demonstration that the MLP approximates the Bayes optimal discriminant function in the large training set limit [20, 6, 10, 17, 1], which is an important theoretical result in the neural-network literature, can easily be extended to this classifier [14].

In a previous work [16], we studied the performances of the FBF network used as a supervised classifier for handwritten digit recognition. It was shown that learning in an FBF network implementing a classifier is faster than in an MLP with a comparable number of parameters. Moreover, the FBF network shows similar generalization performances with respect to the MLP, and this is consistent with the fact that they share the same asymptotical approximation property to the Bayes discriminant function.

3 Structure Identification Problem

When both linguistic descriptions of the classes to be discriminated and numerical data are available, the structure of the FBF network can be initially shaped by the linguistic knowledge and finally refined by training on the numerical data. On the contrary, when only numerical data are available, as in the experiments we report here, the structure identification must be achieved experimentally according to a performance-based criterion.

In theory, when an infinite number of instances are available, the approximation capabilities of a fuzzy system should benefit from using a large number of fuzzy rules [12, 8]. However, in practical applications, a large number of fuzzy rules and the related free parameters, imply storage and computational overhead. Moreover, because of the finite dimension of the

training set, there is the risk of overfitting the data, as usually occurs for neural networks [15].

While the overfitting constitutes an upper bound for the dimension of the FBF network structure, a lower bound, in classification tasks, is given by the semantic phase transition phenomenon, that will be studied in Section 5 in the context of a handwritten digit recognition task. The data and preprocessing of this task are discussed in the next section.

4 A Classification Task: Data Set and Preprocessing

We used a training set and a test set extracted from the NIST-3 data-base [5]. Each set contained 10,000 associative pairs of segmented handwritten characters. The NIST-3 database, distributed on a CD-ROM, contains 313389 characters coded as 128×128 binary matrix images and labeled by the corresponding ASCII codes.

As shown in Figure 1, the preprocessing of character images involved the following steps: a character image was extracted from the CD-ROM and normalized to a 32×32 binary matrix; a low-pass filter was applied in order to remove some small spots and holes from the image; a shear transform was performed on the character image to straighten the axis joining the first upper-left point of the character image to the last lower-right point; the image was then skeletonized by using a thinning algorithm [18]; finally, the character representation was transformed into a 64-element vector, each vector element representing the number of black pixels contained in adjacent 4×4 squares.

It is worth noting that the obtained character representation exhibits sufficient degrees of invariance to both the scale and small image shifts or rotations.

5 Semantic Phase Transition: Experimental Results and Discussion

We have trained several FBF networks, with different numbers of fuzzy rules, by using the training set described in Section 4. The results on both the training and test sets are shown in Figure 2. It can be noticed that FBF networks with fewer than 10 rules yielded poor classification performances, whereas FBF networks with 10 or more rules yielded good classification performances. Moreover, when considering FBF networks with fewer than 10 rules, the addition of a new rule resulted in an improvement in performance of about 10%. On the other hand, for more than 10 rules, improvements were not so significant. Specifically, Tables 1 and 2 show that the training and test error rates, respectively, for systems with fewer than 10 rules concentrated on single classes. For instance [1], FBF_8 cannot recognize class '2' and class '5', while FBF_9 cannot recognize class '6'. The only FBF network that

[1] In the following, the index of "FBF." is the number of rules.

exhibits a slightly different behavior is FBF_4, which recognizes sufficiently well the examples of classes '0', '1', '4', and '7', and is also able to recognize roughly half the examples of class '9'.

The transition of an FBF network from a state of inability to recognize one or more classes (due to the ignorance of the related rules) to a state of sufficient knowledge, was named *semantic phase transition* in [2].

We obtained the semantic phase transition also by pruning the rules from a FBF network with 10 rules, both by removing a single rule at time and by removing the rules sequentially. Specifically, in Table 3, we give the test error rate for each class obtained by FBF_{10} and for $\mathrm{FBF}_9^{(-0)}$, ..., $\mathrm{FBF}_9^{(-9)}$, obtained by removing a single different rule (specified in parenthesis) at a time.

It can be observed that the elimination of a single rule corresponds to the total loss of the capability of the system to classify one specific class, while the performance of the system remains substantially unchanged for the remaining classes.

The same behavior was observed after removing the rules sequentially, as shown in Tables 4 and 5, where the error rate for each class is reported for FBF_{10} and for FBF_{9*}, ... , FBF_{1*}, obtained by removing rules sequentially. It is worth noting that there is a close relationship between the number of rules in an FBF network and the number of classes recognized by the FBF network itself: $10 - \beta$ classes are not recognized at all by the FBF network with β rules, whereas the other classes are well recognized.

An in-depth analysis of the confusion matrices for the above systems shows that the input space is partitioned into β regions, each region being defined by the set of patterns for which one rule is maximally active. This partition of the patterns is evidently demonstrated in Figure 3, where the confusion matrices for FBF_{1*}, FBF_{3*}, FBF_{6*}, and FBF_{9*} are shown. When a rejection criterion is used (e.g., a pattern is rejected if the maximum output is not above a given threshold), each region roughly coincides with the set of patterns of a single class. This can be understood by looking at the confusion matrices obtained for FBF_{6*} by two different rejection thresholds; the two confusion matrices are shown at the bottom of Figure 3. These confusion matrices were obtained by using an independent test set of about 2000 examples. It can be noticed that almost all the patterns of the classes for which there was no specialized rule were rejected.

In conclusion, we have established that the *semantic phase transition* observed for the FBF network can be explained by the *specialization* of each rule in a specific class. Even though it is hard to fully understand how a single rule of the system works, it is very easy to deduce the functional problem solved by each rule. Moreover, we have observed that the set of rules in a system with more than 10 rules can always be partitioned into 10 different subsets, each responsible for the classification of the examples of a specific class.

6 Comparison with MLP: Experimental Results and Discussion

The phenomenon of semantic phase transition observed for the FBF network has not been reported for MLP's in the neural-network literature[2]. Since an FBF network can be considered as a feed-forward connectionist system similar to a three-layer MLP, with as many hidden units as fuzzy rules, we trained four three-layer MLP's with 8, 9, 10, and 12 hidden units, respectively, in order to verify the behavior of an MLP on our data. We used the same training and test sets as used for the FBF network.

As shown in Table 6, for each MLP, the error percentage is almost uniformly distributed over the classes. In general, MLP's with more hidden units are able to obtain better performances than those obtained by MLP's with fewer hidden units. However, we did not observe the semantic phase transition phenomenon, as the error distributions for both MLP's with fewer than 10 hidden units and MLP's with 10 or more hidden units were almost similar, i.e., no specific class was strongly penalized for MLP's with fewer than 10 hidden units.

Then, we can conclude that, whereas in an FBF network each rule specializes in a particular output component, in an MLP all hidden units, in accordance with the globality of the sigmoidal activation functions, contribute to the recognition of all classes, and no hidden unit specializes in one particular output class. As a consequence, without any specific knowledge on the classification task, it is impossible to determine a priori the proper size of the hidden layer of an MLP, or at least to determine a lower bound to it.

7 Conclusions

In this paper, we have studied the behavior of the FBF network [9, 25] based on height method defuzzifier, product-inference rule, singleton fuzzifier, and Gaussian membership function. We have experimentally shown that, when only numerical data are available, the choice of a proper structure for the system is bounded on the number of fuzzy rules.

On one hand, the theory states that, in the large training set limit, the approximation capabilities of a fuzzy system should benefit from using a large number of fuzzy rules. For real problems, however, the finite dimension of the training set gives rise to overfitting problems.

On the other hand, 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 to be discriminated. Specifically, we have demonstrated that an FBF network with fewer rules than classes to be discriminated is unable to obtain a reasonable classification performance because of the inability of the system to recognize whole classes, i.e., each rule specializes

[2] A comparable behavior can be elicited from an MLP only by adding to the error function a special term constraining the hidden representations.

in a specific class. When the number of rules is increased up to the number of classes to be discriminated, a sharp increase in the performance is observed (*semantic phase transition*).

We can conclude that, in practical applications, an FBF network exhibits the same overfitting problems as a neural network. Moreover, even if an FBF network is organized as a supervised feed-forward network, its behavior is closer to a competitive model showing a strong specialization of the fuzzy rules.

Acknowledgments

This work was supported by grants from CNR-Progetto Strategico Reti Neurali, GNCB-CNR, INFM, and MURST. We thank Renato Caviglia for helpful discussions.

References

[1] Barnard, E., Kanaya, F., Miyake, S. "Comments on 'Bayes statistical behavior and valid generalization of pattern classifying neural networks' (with reply)." *IEEE Transactions on Neural Networks*, 3, 1992, pp. 1026–7.

[2] Casalino, F., Masulli, F., Sperduti, A., and Vannucci, F. "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, IEEE, 1994, pp. 808–812.

[3] Caviglia, R. *Soft-computing methods for time series forecasting.* (in Italian), Laurea Thesis in Computer Science, University of Genoa, Genova (Italy), 1994.

[4] Cybenko, G. "Approximation by superpositions of a sigmoidal function." *Mathematics of Control, Signals, and Systems*, 2 , 1989, pp. 303–314.

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

[6] Hampshire, J., and Pearlmutter, B. "Equivalence proofs for multi-layer perceptron classifiers and the Bayesian discriminant function." In D.S. Touretzky, G. Hinton, and T. Sejnowski, editors, *Connectionist Models: Proceedings of the 1990 Summer school*, Morgan Kaufmann, San Mateo, 1990, pp. 13–19.

[7] Jang, J.S.R. "ANFIS: Adaptive-network-based fuzzy inference system." *IEEE Trans. on Systems, Man, and Cybernetics*, 23, 1993, pp. 655–684.

[8] Jou, C.C. "On the mapping capabilities of fuzzy inference systems." In *IJCNN International Joint Conference on Neural Networks*, volume 2, IEEE, 1992, pp. 703–713.

[9] Jou, C.C. "Comparing learning performance of neural networks and fuzzy systems." In *IEEE International Conference on Fuzzy Systems*, IEEE, 1993 , pp. 1028–1033.

[10] Kanaya, F., and Miyake, S. "Bayes statistical behavior and valid generalization of pattern classifying neural networks." *IEEE Transactions on Neural Networks*, 2, 1991, pp. 471–475.

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

[12] B. Kosko. "Fuzzy systems as universal approximators". *IEEE Transactions on Computers*, 43, 1994, pp. 1329–33.

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

[14] Masulli, F. "Bayesian classification by feedforward connectionist systems." In F. Masulli, P. G. Morasso, and A. Schenone, editors, *Neural Networks in Biomedicine*, World Scientific, 1994, pp. 145–162.

[15] Masulli, F., Casalino, F., Caviglia, R., and Papa, L. "Comparison of statistical methods and fuzzy systems in atmospheric pressure wave prediction." In S.K Roger and D.W. Ruck, editors, *Applications and Science of Artificial Neural Networks - SPIE Proceedings Series*, volume 2492, SPIE, 1995, pp. 1050–1061.

[16] Masulli, F., Casalino,F., and Vannucci, F. "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*, Springer, 1994, pp. 189–192.

[17] Miyake, S., and Kanaya, F. "A neural network approach to a Bayesian statistical decision problem." *IEEE Transactions on Neural Networks*, 2, 1991, pp. 538–540.

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

[19] Poggio, T., and Girosi, F. "Regularization algorithms for learning that are equivalent to multilayer networks." *Science*, 247, 1990, pp. 978–982.

[20] Ruck, D.W., Rogers, S.K., Kabrisky, M., Oxley, M.E., and Suther, B.W. "The multilayer perceptron as an approximation to a Bayes optimal discriminant function." *IEEE Transactions on Neural Networks*, 1, 1990, pp. 296–298.

[21] Rudin, W. *Principles of Mathematical Analysis.* McGraw-Hill, Inc., 1960.

[22] Rumelhart, D.E., Hinton, G.E., and Williams, R.J. "Learning internal representations by error propagation." In D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing*, volume 1, chapter 8, MIT Press, 1986, pp. 318–362.

[23] Wang, L. X. "Fuzzy systems are universal approximators", In *IJCNN International Joint Conference on Fuzzy Systems*, IEEE, 1992, pp. 1163–1170.

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

[25] Wang, L. X. *Adaptive Fuzzy Systems and Control.* Prentice Hall, 1994.

Captions for Tables

Table 1: Training set error rates by ten different FBF networks, with 1 to 10 rules and trained separately.

Table 2: Test set error rates by ten different FBF networks, with 1 to 10 rules and trained separately.

Table 3: Test set error rates by FBF_{10} and by ten different FBF networks obtained by pruning a different single rule from FBF_{10}.

Table 4: Training set error rates by ten different FBF networks, with 1 to 10 rules, obtained by pruning FBF_{10} sequentially.

Table 5: Test set error rates by ten different FBF networks, with 1 to 10 rules, obtained by pruning FBF_{10} sequentially.

Table 6: Training set error rates by four different MLP's (with 8, 9, 10 and 12 hidden units).

Captions for Figures

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

Figure 2: Success rates on training and test sets obtained by FBF networks ranging from 8 to 64 rules.

Figure 3: Confusion matrices related to the training and test sets, for FBF_{1*}, FBF_{3*}, FBF_{6*}, and FBF_{9*}. Each matrix element represents the percentage of examples labeled C_i that are recognized in the class R_i, and is given by the area of the black squares. At the bottom of the figure, the confusion matrices obtained for FBF_{6*} by two different rejection thresholds (.85 and .95) are shown; R is the rejection class.

	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Examples for class	1052	1134	966	1059	976	842	948	1052	978	1002
FBF_1	100.00	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
FBF_2	0.12	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
FBF_3	0.48	0.00	100.00	99.91	100.00	100.00	100.00	82.00	100.00	2.59
FBF_4	0.86	0.18	100.00	96.51	6.72	100.00	100.00	5.32	99.69	51.90
FBF_5	1.52	1.76	100.00	4.72	97.00	100.00	6.55	83.75	99.59	3.39
FBF_6	1.24	2.20	100.00	3.40	94.73	100.00	4.54	3.52	100.00	6.09
FBF_7	1.52	2.03	100.00	4.06	2.48	100.00	3.69	2.85	8.49	94.71
FBF_8	1.71	1.41	100.00	4.44	3.52	100.00	4.01	2.57	9.82	3.59
FBF_9	1.52	1.50	10.56	4.15	3.10	12.35	100.00	2.57	7.46	3.79
FBF_{10}	1.81	1.59	6.11	4.44	3.83	11.63	4.54	2.47	8.69	3.59

Table 1: Training set error rates by ten different FBF networks, with 1 to 10 rules and trained separately.

	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Examples for class	1019	1127	982	1049	944	870	980	1035	1011	983
FBF_1	100.00	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
FBF_2	0.59	0.18	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
FBF_3	0.69	0.09	100.00	100.00	100.00	100.00	100.00	79.72	100.00	2.34
FBF_4	1.37	0.62	100.00	96.00	5.83	100.00	100.00	4.54	99.41	52.19
FBF_5	2.75	2.49	100.00	6.29	93.96	100.00	5.92	81.26	99.31	3.26
FBF_6	2.65	2.75	100.00	5.53	94.07	100.00	4.18	3.38	100.00	5.90
FBF_7	3.63	2.48	100.00	6.58	3.92	100.00	3.78	4.44	9.10	95.22
FBF_8	4.12	2.40	100.00	7.24	6.36	100.00	4.39	4.64	10.68	4.78
FBF_9	3.73	2.31	14.97	7.91	6.78	19.08	100.00	4.54	10.98	5.49
FBF_{10}	4.51	2.30	12.63	8.58	8.16	18.85	6.94	6.09	10.68	4.58

Table 2: Test set error rates by ten different FBF networks, with 1 to 10 rules and trained separately.

	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Examples for class	1052	1134	966	1059	976	842	948	1052	978	1002
FBF_{10}	1.81	1.59	6.11	4.44	3.83	11.63	4.54	2.47	8.69	3.59
$FBF_9^{(-0)}$	1.71	1.50	6.00	4.00	3.83	*100.00*	4.11	2.47	8.08	3.49
$FBF_9^{(-1)}$	1.91	1.59	5.69	*100.00*	3.83	10.92	4.54	2.47	8.49	3.49
$FBF_9^{(-2)}$	1.81	1.50	5.49	4.44	3.83	11.63	4.54	*100.00*	8.49	3.30
$FBF_9^{(-3)}$	1.71	1.58	*100.00*	4.25	3.83	11.52	4.43	2.19	7.98	3.49
$FBF_9^{(-4)}$	*100.00*	1.59	6.00	4.34	3.62	11.63	4.54	2.38	8.49	3.49
$FBF_9^{(-5)}$	1.71	1.41	5.69	4.44	3.41	11.40	*100.00*	2.47	8.82	3.59
$FBF_9^{(-6)}$	1.81	*100.00*	5.90	4.34	3.72	11.04	3.38	2.38	8.59	3.39
$FBF_9^{(-7)}$	1.81	1.59	6.11	4.34	2.79	11.40	4.53	2.28	6.34	*100.00*
$FBF_9^{(-8)}$	1.41	1.50	6.11	4.44	*100.00*	11.63	4.43	2.47	8.28	3.29
$FBF_9^{(-9)}$	1.80	1.59	5.07	4.34	3.62	11.52	4.54	2.47	*100.00*	3.29

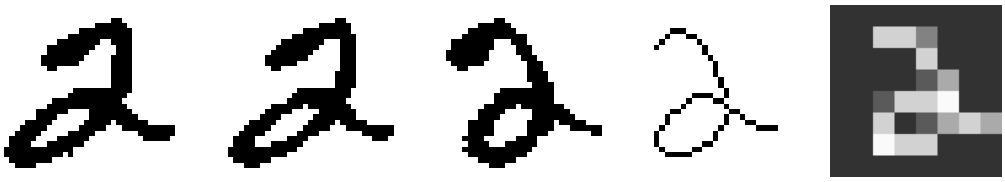
Table 3: Test set error rates by FBF_{10} and by ten different FBF networks obtained by pruning a different single rule from FBF_{10}.

	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Examples for class	1052	1134	966	1059	976	842	948	1052	978	1002
FBF_{10}	1.81	1.59	6.11	4.44	3.83	11.63	4.54	2.47	8.69	3.59
FBF_{1*}	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	0.00	100.00
FBF_{2*}	100.00	100.00	100.00	100.00	0.83	100.00	100.00	100.00	1.64	100.00
FBF_{3*}	100.00	100.00	100.00	100.00	1.96	100.00	100.00	100.00	5.73	2.30
FBF_{4*}	100.00	1.23	100.00	100.00	2.38	100.00	100.00	100.00	6.03	2.69
FBF_{5*}	100.00	1.41	100.00	100.00	3.31	100.00	4.00	100.00	6.75	2.69
FBF_{6*}	1.62	1.41	100.00	100.00	3.83	100.00	4.00	100.00	6.95	2.89
FBF_{7*}	1.71	1.41	4.97	100.00	3.83	100.00	4.11	100.00	7.67	3.09
FBF_{8*}	1.71	1.50	5.59	100.00	3.83	100.00	4.11	2.47	7.87	3.39
FBF_{9*}	1.71	1.50	6.00	3.97	3.83	100.00	4.11	2.47	8.08	3.49

Table 4: Training set error rates by ten different FBF networks, with 1 to 10 rules, obtained by pruning FBF_{10} sequentially.

	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Examples for class	1019	1127	982	1049	944	870	980	1035	1011	983
FBF_{10}	4.51	2.31	12.63	8.58	8.16	1 8.85	6.94	6.09	10.68	4.57
FBF_{1*}	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	0.00	100.00
FBF_{2*}	100.00	100.00	100.00	100.00	3.18	100.00	100.00	100.00	1.88	100.00
FBF_{3*}	100.00	100.00	100.00	100.00	5.51	100.00	100.00	100.00	5.84	2.85
FBF_{4*}	100.00	1.60	100.00	100.00	5.51	100.00	100.00	100.00	6.03	3.05
FBF_{5*}	100.00	1.69	100.00	100.00	6.46	100.00	4.16	100.00	7.42	3.05
FBF_{6*}	3.63	1.77	100.00	100.00	7.20	100.00	3.78	100.00	8.01	3.05
FBF_{7*}	4.02	1.95	10.39	100.00	7.94	100.00	5.41	100.00	9.10	3.15
FBF_{8*}	4.02	2.22	11.20	100.00	7.94	100.00	5.41	5.99	9.20	4.17
FBF_{9*}	4.02	2.31	12.53	7.34	7.94	100.00	4.51	6.09	9.30	4.17

Table 5: Test set error rates by ten different FBF networks, with 1 to 10 rules, obtained by pruning FBF_{10} sequentially.



(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).

	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Examples for class	1052	1134	966	1059	976	842	948	1052	978	1002
MLP_8	2.76	3.53	8.18	5.76	4.96	8.67	2.53	2.66	8.18	3.69
MLP_9	2.57	2.38	6.42	4.53	3.52	6.41	2.32	2.00	6.95	2.99
MLP_{10}	2.76	3.26	7.45	6.04	6.00	6.65	2.64	2.95	7.16	3.69
MLP_{12}	3.23	2.82	8.18	5.48	4.45	6.89	2.53	4.18	8.49	4.49

Table 6: Training set error rates by four different MLP's (with 8, 9, 10 and 12 hidden units).

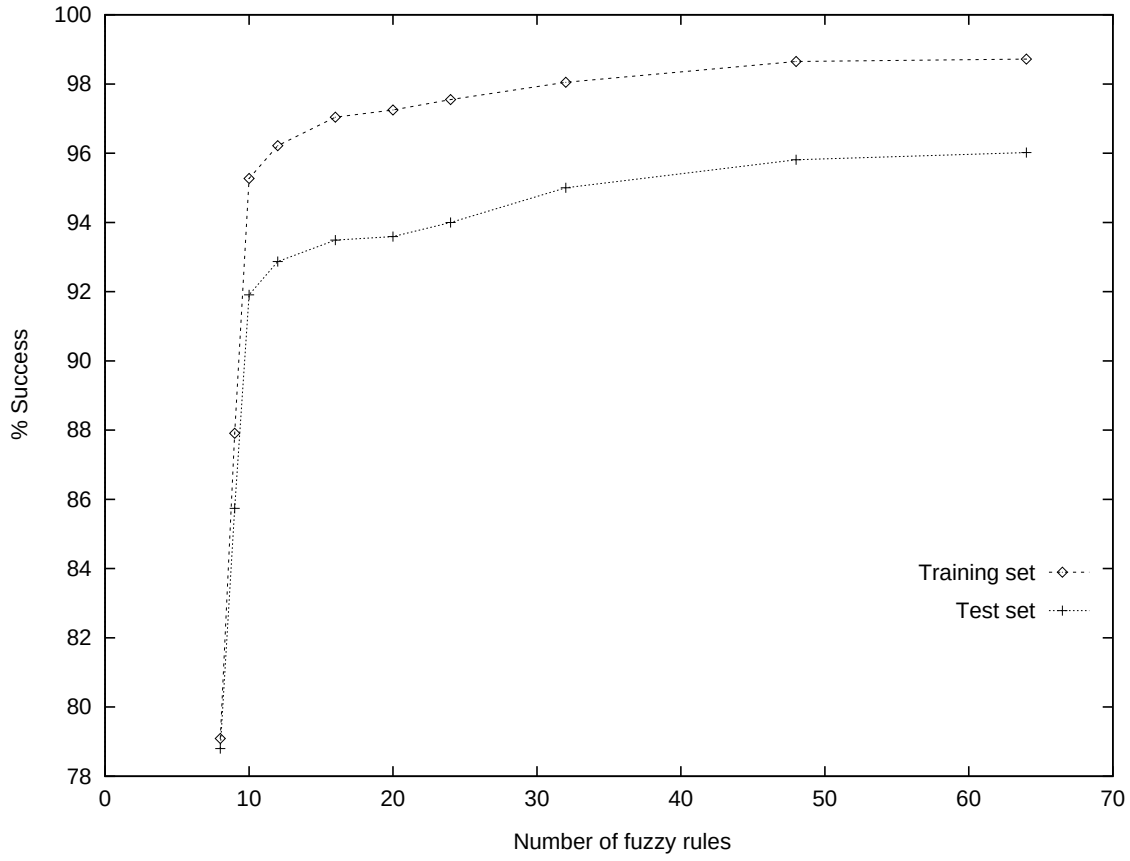


Figure 2: Success rates on training and test sets obtained by FBF networks ranging from 8 to 64 rules.

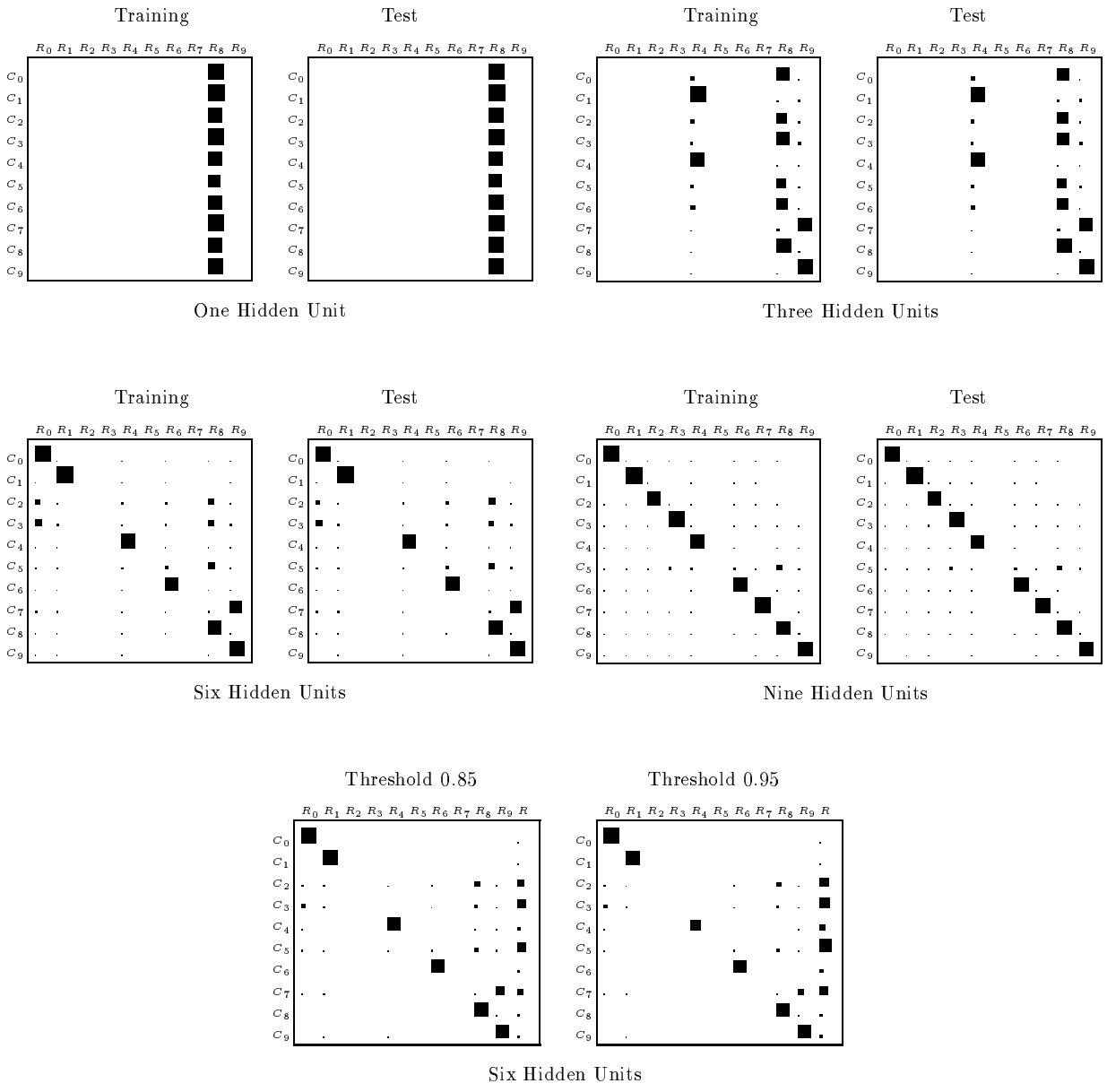


Figure 3: Confusion matrices related to the training and test sets, for FBF_{1*}, FBF_{3*}, FBF_{6*}, and FBF_{9*}. Each matrix element represents the percentage of examples labeled C_i that are recognized in the class R_i, and is given by the area of the black squares. At the bottom of the figure, the confusion matrices obtained for FBF_{6*} by two different rejection thresholds (.85 and .95) are shown; R is the rejection class.