Training without data: Knowledge Insertion into RBF Neural Networks

Ken McGarry and Stefan Wermter
University of Sunderland
Department of Computing and Technology
David Goldman Informatics Centre, St Peters Campus, St Peters Way, SR6 ODD
ken.mcgarry@sunderland.ac.uk, www.his.sunderland.ac.uk

Abstract

A major problem when developing neural networks for machine diagnostics situations is that no data or very little data is available for training on fault conditions. However, the domain expert often has a good idea of what to expect in terms of input and output parameter values. If the expert can express these relationships in the form of rules, this would provide a resource too valuable to ignore. Fuzzy logic is used to handle the imprecision and vagueness of natural language and provides this additional advantage to a system. This paper investigates the development of a novel knowledge insertion algorithm that explores the benefits of pre-structuring RBF neural networks by using prior fuzzy domain knowledge and previous training experiences. Pre-structuring is accomplished by using fuzzy rules gained from a domain expert and using them to modify existing Radial Basis Function (RBF) networks. The benefits and novel achievements of this work enable RBF neural networks to be trained without actual data but to rely on input to output mappings defined through expert knowledge.

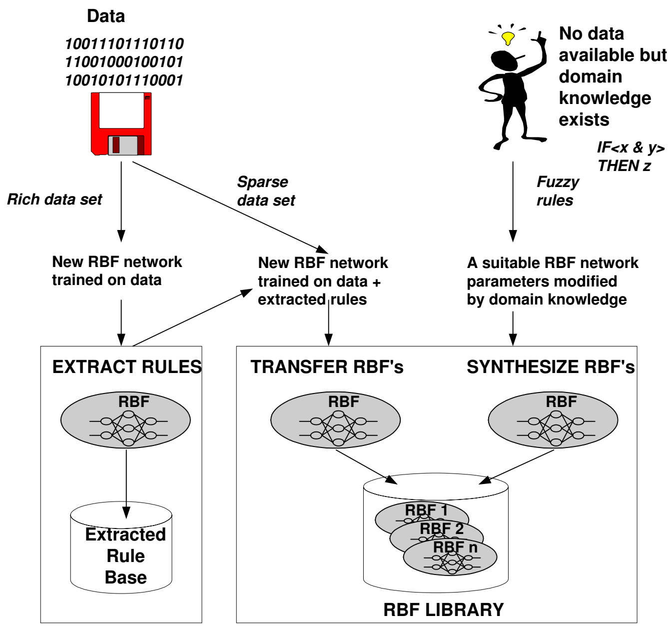

Figure 1: Overview of Architectural Transformations

1 Introduction

Radial Basis Function (RBF) neural networks are a form of local learning and are an efficient alternative to Multi-layer Perceptron (MLP) neural networks. In this paper we investigate how symbolic knowledge can be inserted into an RBF network, the advantage being to capitalize on those situations where no training data exists but an domain expert may be able to formulate IF..THEN rules. Figure 1 presents an overview of a much larger system that was developed, the work on rule extraction and knowledge transfer described elsewhere. It is the module related to the synthesis of RBF networks that is the focus of this paper.

Neural networks, like any other inductive learning algorithm such as decision trees and clustering techniques are data driven. Without a sufficient supply of training data these algorithms cannot be used to produce a reliable model of the problem domain. In such cases, the usual solution to these problems range from duplicating data with additional noise added or to simulate the data. The extraction of knowledge in the form of rules has been successfully explored before on RBF networks using the *hREX* algorithm [McGarry *et al.*, 2001]. This work inspired the authors to develop knowledge synthesis or knowledge insertion by manipulating the RBF network parameters but information flow/extraction was in the opposite direction.

Domain experts often express their knowledge in vague and imprecise linguistic terms. It is therefore a natural step to use the ability of fuzzy sets and fuzzy logic to model this imprecision. Fuzzy sets and rules created by the knowledge elicitated from a domain expert are used to manipulate the parameters of selected RBF hidden units. Knowledge synthesis is a technique to be used in those situations where data for training is unavailable but specific knowledge about the application is at hand. Converting the fuzzy rules into RBF network format avoids the integration issue of assigning confidence factors when interpreting conflicting module outputs. Therefore, the local representation characteristics of the RBF networks are used to integrate hidden units placed by data driven learning with artificially synthesized hidden units.

The remainder of this paper is structured as follows: section two describes the Radial Basis Function neural network, and the how certain functional equivalences to fuzzy systems are exploited. Section three describes the work on knowledge synthesis for RBF networks for improved performance. Section four outlines the details of the fuzzy system employed and how it integrates the domain rules with the RBF network. Section five discusses the experimental results. Section six provides a brief overview of the related work and finally section seven presents the conclusions and areas for future work.

2 Radial Basis Function Networks

Radial basis function (RBF) neural networks are a model that has functional similarities found in many biological neurons. In biological nervous systems certain cells are responsive to a narrow range of input stimuli, for example in the ear there are cochlear stereocilla cells which are locally tuned to particular frequencies of sound [Moody and Darken, 1989]. The RBF network consists of a feedforward architecture with an input layer, a hidden layer of RBF "pattern" units and an output layer of linear units. The input layer simply transfers the input vector to the hidden units, which form a localized response to the input pattern. The activation levels of the output units provide an indication of the nearness of the input vector to the classes. Learning is normally undertaken as a two-stage process.

The radial basis functions in the hidden layer are implemented by kernel functions, which operate over a localized area of input space. The effective range of the kernels is determined by the values allocated to the centre and width of the radial basis function. The Gaussian function has a response characteristic determined by equation 1.

$$Z_j(x) = exp\left(-\frac{||x-\mu||^2}{\sigma_j^2}\right) \tag{1}$$

where: μ = n-dimensional parameter vector, σ = width of receptive field, and *x* = input vector.

The response of the output units is calculated using equation 2.

$$\sum_{j=l}^{J} W_{lj} Z_j(x) \tag{2}$$

where: *W* = weight matrix, *Z* = hidden unit activations and *x* = input vector.

RBF networks are an appropriate choice for both classification tasks and function approximation. It is interesting to note that already Jang [Jang and Sun, 1993] discovered certain functional similarities exist between RBF networks and fuzzy systems. Jang identified five criteria that are necessary for the two techniques to become functionally equivalent.

1. The number of RBF units must be equal to the number of fuzzy rules.
2. The output of each fuzzy rule is a constant.
3. The membership functions within each rule are gaussians with the same widths.
4. The t-norm operator computes the firing strength of each rule through multiplication.
5. The RBF network and the fuzzy inference system both use either normalized or unnormalized output calculations.

The main advantage of the functional equivalence relationship is the ability of using one model's learning rules for the other and vice-versa. Other Fuzzy-RBF researchers [Hunt *et al.*, 1996; Halgamuge, 1997] have extended Jang's work to produce hybrid learning models. Fuzzy models are generally robust when faced with noisy and missing data and are more comprehensible than a purely neural network approach [Sun, 1994].

3 Knowledge Synthesis

Knowledge synthesis is a technique intended for those situations in which no actual training data is available but some form of domain knowledge is at hand. The experts knowledge is encoded as fuzzy sets and rules which are used to synthesize new hidden unit parameters for incorporation into a new or existing network. The fuzzy rules describe a set of output classes and the possible input values denoting their characteristics (based on the experts opinions). The objective of converting from fuzzy rules to RBF networks is to have the knowledge in a consistent format. It would be possible to have the domain knowledge in the form of a stand-alone fuzzy module, interacting with the RBF based system in some loosely or tightly coupled protocol. However, by converting the fuzzy rules into an RBF architecture they can be subjected to further analysis by rule extraction and it also avoids hybrid system integration issues.

The RBF hidden layer parameters (spread and centre values) can be converted from fuzzy sets by the rule defined by [Jang and Sun, 1993]:

$$\mu_{A_1}(x_1) = exp[-\frac{(x_1 - c_{A_1})^2}{\sigma_1^2}] \tag{3}$$

Where:$\mu_{A_1}(x_1)$ is the fuzzy linguistic labels containing the domain knowledge, c is the centre of the receptive field i.e. the RBF function, σ_1^2 is the RBF spread. Modification of the RBF output layer parameters was accomplished by use of the pseudoinverse matrix X^P [Kubat, 1998].

$$W = (X^T X)^{-1} X^T C = X^P C \tag{4}$$

Where: C is the classification matrix containing class labels, W is the hidden-to-output unit weight matrix, X is the matrix containing the converted information from fuzzy sets (gained from eqn 3).

3.1 The Problem Domain

An industrial machine vibration data set was chosen because of the availability of several domain rules relating spectral data to fault conditions. This knowledge is available in text books, journals and rules taken from a large European collaborative project.

A common problem encountered with many diagnostic applications is the lack of data for certain system conditions especially fault situations. The diagnosis data obtained for the experimental work described in this paper overcame such problems by using a mixture of simulated and test rig data. This data enabled an experiment to be conducted which assumed that data collected at low motor speeds was available but not at high speeds. At high motor speeds the values of several input features can be expected to change (increase or decrease) also the severity of the faults may appear different. However, by using vibration theory heuristics it was possible to predict how certain parameters and fault conditions would behave at higher speeds. The following method was used for the vibration data experiment:

3.2 Methodology

1. The vibration data was divided into two groups, i. motor speed classed as low and ii. motor speed classed as high.
2. An RBF network was trained on the low speed data and the accuracy was examined with a separate low speed test data.
3. The RBF network was tested with the high speed data (i.e. the data it was *not* trained on) and the accuracy was noted.
4. The domain rules were converted into fuzzy sets and fuzzy rules.
5. The *h*REX algorithm was used to assign hidden units to the output classes.
6. For each of the three fault classes considered the hidden units assigned to them were duplicated and modified using the fuzzy inferencing system.
7. The modified RBF network had its hidden-to-output weights and biases recalculated.
8. The modified network was re-tested on the high speed data and the accuracy noted.

The vibration data was divided into two sets based on the speed of the motor: low speed data consisting of examples with motor speeds of 500-1000 RPM (rotations per minute) and high speed data containing data from motor speeds of 1500-2000 RPM. However, both sets of data still consisted of 11 input features and eight output classes. An RBF network was trained on the low speed data and its accuracy was observed on both low speed and high speed test data. The results are shown in figure 2.

The RBF network trained on low speed data has an accuracy 90% on the low speed test set. Unfortunately, the accuracy falls dramatically to 51.3% when introduced to the high speed data. This is a common phenomena that RBF neural networks can generate well to new patterns only if they stem from the same distribution of input patterns. If the task is very different they do not generalise well. The unreliability of such a classifier would prevent it from being deployed in any application running at high speed. Therefore, a substantial increase in accuracy is required for knowledge synthesis to be of practical use.

Classification rate: 90%

109	0	5	2	0	0	0	5
11	31	0	0	0	0	0	0
0	0	53	0	0	0	0	0
2	0	0	24	0	0	0	1
0	0	0	0	38	0	0	0
0	0	0	0	0	15	2	0
0	0	0	0	0	5	27	0
2	0	0	1	0	0	0	31

Original network accuracy on low speed data

Classification rate: 51.3%

594	6	15	9	0	2	16	25
99	0	0	0	0	0	0	0
155	0	0	0	0	0	37	0
104	0	0	0	0	0	0	0
95	0	0	0	0	0	0	0
46	0	0	1	12	33	0	2
16	0	0	1	0	1	78	0
36	0	0	1	0	0	24	38

Original network accuracy on high speed data

Figure 2: Confusion matrix showing accuracy of the original RBF network. The numbers represent test cases and those lying on the diagonal have been classified accurately, while those off the diagonal have been misclassified. The accuracy of the RBF network trained on low speed data is 90.0%, but when introduced to test data with high speed characteristics, accuracy falls to 51.3%. High speed training data is not available, thus motivating the need to integrate some form of knowledge insertion.

Although the original RBF uses 11 input parameters only three parameters were used in the experimental work. This is because the majority of heuristic rules use only these (rpm1, rpm2 and rpm3) parameters because most faults can be identified by them. In addition, only three machine conditions were considered: unbalance, misalignment and looseness.

4 Defining the Fuzzy System

The knowledge for the fuzzy system was obtained from a domain expert and the available machine fault diagnosis literature. The fuzzy rules, sets and membership functions were manually developed through a process of trial and error.

4.1 Defining the fuzzy sets

A number of criteria were under consideration for the construction of the fuzzy sets.

- Type of membership function (Gaussian, triangular etc).
- Number of membership functions per set.
- Number of rules to required to cover the intended input/output space.
- Defuzzification method used.

The type of membership function to be used proved not to be critical, as the set for the motor speed uses triangular but similar results were achieved for Gaussian. As long as there is some overlap between the membership functions then the actual type of function does not appear to affect the results to any great extent (this is true for the vibration domain).

The motor speed fuzzy set was used as an input to the inferencing system. Triangular functions were used with the apex of the triangle at the mid-value of the set. The "slow" membership function is present only for completeness as it did not participate in the fuzzy inferencing process. Figure 3 shows the membership functions for the fault severity. This set was used as an input to the system and the values for the membership were determined by the domain experts belief as how much a fault has to be present (percentage) to constitute a problem. In figure 3 the looseness fault can be seen as a major problem for the motor since only a small amount of looseness is required to be classed as a serious fault. Looseness refers to the amount of physical movement experienced by the motor and usually indicates that the mounting bolts are not secure.

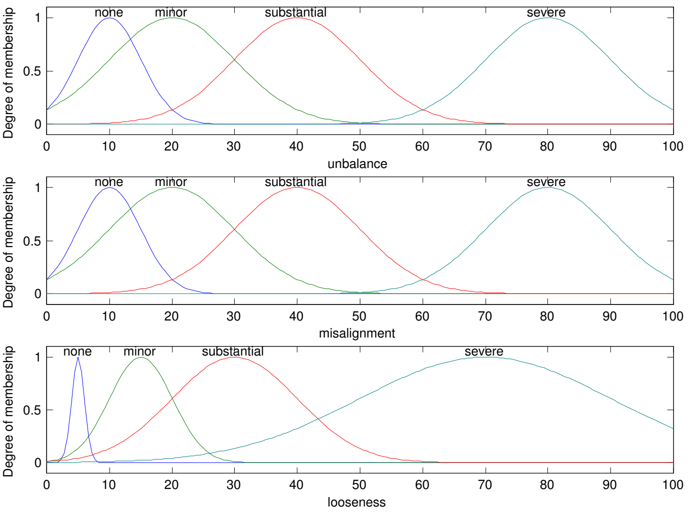

Figure 3: Fuzzy sets defining the fault severity

Figure 4 shows the membership functions for the rpm parameters. These fuzzy sets were used as outputs by the system and the values for the rpm parameters were calculated by the inferencing system. Those values were then used to modify the centre positions of the new RBF hidden units. The membership values were mainly based on averages of both simulated and test-rig high speed data. The diagnostic system used rules based on testing the ratios of the rpm parameters as opposed to absolute values. Some of these rules provided useful information that was incorporated into the fuzzy rules to determine the rpm parameter values.

4.2 Defining the fuzzy rules

Figure 5 shows 10 of the 18 rules for the three fault conditions: unbalance, misalignment and looseness. Rule 19 was present to prevent the fuzzy system from making a decision when no fault conditions were present. Each fuzzy rule consisted of an antecedent and consequent. The antecedent was

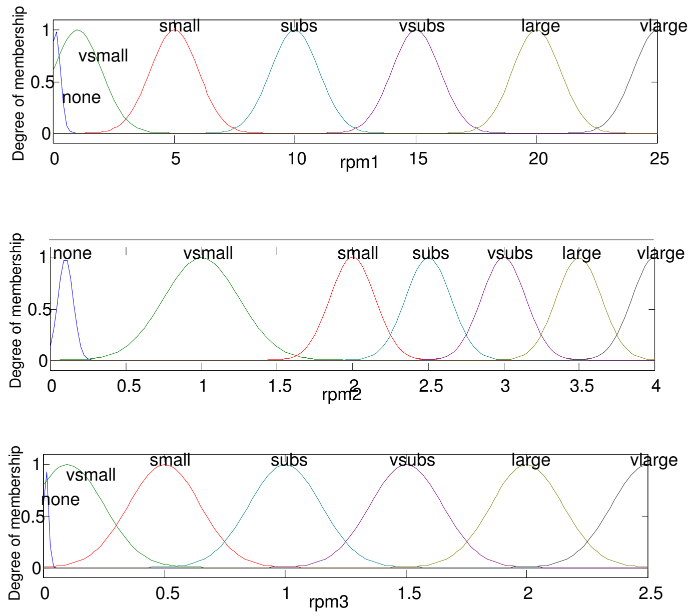

Figure 4: Fuzzy sets defining the parameter modifications

quite simple compared with the rules extracted from the RBF networks as there were only two input features i.e. the motor speed and fault severity. The consequent was more complex as it consisted of values assigned to the three output parameters.

The structure of the rules were almost identical to the reasoning of the domain expert. In addition to comprehensibility there is also the added advantage of system maintainability. Adding further fuzzy rules to cover extra conditions could be easily made but without the problems of "side-effects" that can be encountered in non-fuzzy systems i.e. adding extra rules may cause unanticipated logic errors. Also the coding complexity in non-fuzzy systems can be greatly increased. This was avoided to a great extent with the fuzzy system because a lot of the knowledge was encoded into the fuzzy sets. Therefore, if required the fuzzy sets could be re-calibrated/altered to take advantage of new situations without recoding the rules.

4.3 Fuzzy Inferencing System

The inferencing method used is the Mamdani method which is the most commonly used method for this type of fuzzy system [Mamdani and Baakini, 1974]. It enabled multiple fuzzy sets to be used as outputs (i.e. in the rule consequents). These fuzzy sets are eventually used to modify the RBF internal parameters (after defuzzification).

The system operates by combining the consequents from each rule through the aggregation operator and returning a defuzzified fuzzy set to provide the system output. The fuzzy Mamdani model is composed of rule sets of IF..THEN type rules [Mitra and Hayashi, 2000].

In order to identify which particular hidden RBF units to

Rule 1: If (MSpeed is fast) and (unbalance is minor)
then (rpm1 is vsmall)(rpm2 is none)(rpm3 is none)

Rule 2: If (MSpeed is fast) and (unbalance is substantial)
then (rpm1 is small)(rpm2 is vsmall)(rpm3 is vsmall)

Rule 3: If (MSpeed is fast) and (unbalance is severe)
then (rpm1 is subs)(rpm2 is subs)(rpm3 is subs)

Rule 4: If (MSpeed is vfast) and (unbalance is minor)
then (rpm1 is vsubs)(rpm2 is vsubs)(rpm3 is vsubs)

Rule 5: If (MSpeed is vfast) and (unbalance is substantial)
then (rpm1 is large)(rpm2 is large)(rpm3 is large)

Rule 6: If (MSpeed is vfast) and (unbalance is severe)
then (rpm1 is vlarge)(rpm2 is vlarge)(rpm3 is vlarge)

Rule 7: If (MSpeed is fast) and (misalignment is minor)
then (rpm1 is small)(rpm2 is vsmall)(rpm3 is none)

Rule 8: If (MSpeed is fast) and (misalignment is substantial)
then (rpm1 is small)(rpm2 is vsmall)(rpm3 is vsmall)

Rule 9: If (MSpeed is fast) and (misalignment is severe)
then (rpm1 is subs)(rpm2 is subs)(rpm3 is subs)

Rule 10: If (MSpeed is vfast) and (misalignment is minor)
then (rpm1 is subs)(rpm2 is subs)(rpm3 is subs)

Figure 5: Fuzzy domain rules for machine vibration domain, where: *looseness*, *misalignment*, *unbalance* are the faults that can be encountered by the machines, *minor, substantial severe,* describe the extent of the fault, *rpm1,rpm2,rpm3* are the parameters that indicate the presence or absence of the faults, *none, small, vsmall, subs* are the amount of change required to the parameters.

modify, the *h*REX algorithm was implemented [McGarry *et al.*, 2001] and was run on the RBF network which produced a list of hidden units contributing to the identification of the three fault classes. Table 1 details the hidden unit allocations for the classes.

Having identified the relevant hidden units, their associated centres, spreads and hidden-to-output unit weights were modified by the fuzzy inferencing system according to the values of the fuzzy sets.

5 Analysis of Knowledge Synthesis Results

The results are shown in figure 6. Overall, an improvement of 25% was made with unbalance gaining 88 correct, misalignment gaining 67 and looseness gaining 30 correct classifications. The original RBF network had an overall tendency to misclassify most examples as “ok” but now the domain rules enabled tighter bounds to be placed around the “ok” category.

Table 1: Identification of hidden units with output classes

Fault Class	**hidden units responsive to fault class**
Unbalance	7, 9, 12, 13, 15, 23, 28, 31
Misalignment	3, 8, 15
Looseness	13, 14, 16, 21, 32

Classification rate: 51.3%

594	6	15	9	0	2	16	25
99	0	0	0	0	0	0	0
155	0	0	0	0	0	37	0
104	0	0	0	0	0	0	0
95	0	0	0	0	0	0	0
46	0	0	1	12	33	0	2
16	0	0	1	0	1	78	0
36	0	0	1	0	0	24	38

Original network accuracy on high speed data

Classification rate: 75.0%

594	2	15	9	0	2	16	25
15	88	0	0	0	0	0	0
88	0	67	0	0	0	37	0
74	0	0	30	0	0	0	0
95	0	0	0	0	0	0	0
46	0	0	1	12	33	0	2
16	0	0	1	0	1	78	0
36	0	0	1	0	0	24	38

Modified network accuracy on high speed data

Figure 6: Confusion matrix showing accuracy of the original RBF network compared with synthesized RBF. The numbers represent test cases and those lying on the diagonal have been classified accurately, while those off the diagonal have been misclassified. The original network has an accuracy of 51.3% on the high speed data but when it is modified by inserting domain rules that are characteristic of the nature of high speed data, the accuracy goes up to 75.0%.

The effects of the large number of shared hidden units (33%) within the network are still an open issue. Theoretically, since each shared hidden unit is duplicated and modified for each class there should be no undue interference effects. However, with additional rules these results could be improved upon but obtaining such rules would be difficult as the inter-parameter relationships are complex.

The existing rules cover a relatively simple input to output space mapping that was common to most machine behavior (generic) and was easy to produce from the knowledge acquisition process. It would be possible to conduct a statistical analysis of the high speed data for particular relationships and then to produce rules which could be used on different types of machines i.e. the analysis might have proved that certain relationships exist between low and high speed data on machine “X” and then to have applied these rules to machine “Y”. The difficulty of this approach is that of modelling the specific data too closely and not the underlying relationships (if they exist).

6 Related Work on Knowledge Synthesis

Previous work involving assigning prior knowledge into locally responsive units used a Bayesian framework which enabled the incorporation of an inductive bias [Roscheisen *et al.*, 1994]. Other approaches have considered the problem in terms of inserting symbolic rules into a RBF network [Andrews and Geva, 1996; Tresp *et al.*, 1997].

In the past researchers have attempted to incorporate domain knowledge in the form of fuzzy rules within a neural network. Jin and Sendoff [Jin and Sendhoff, 1999] use fuzzy rules in the form of hints as described by Abu-Mostafa [Abu-Mostafa, 1993]. The "hint" is expressed as a fuzzy rule that describes qualitively the function to be learned and is used as a regularization term by the learning algorithm. Narazaki describes the use of fuzzy rules to manipulate the classification boundaries of the hidden units in a MLP network [Narazaki, 1996]. Casalino developed a fuzzy basis function (FBF) network that has some similarity of operation with an RBF network [Casalino *et al.*, 1998]. However, the main conclusion of Casalino work was to identify that the FBF characteristics were closer to a competitive learning model than a feedforward. The work by Kishore [Kishore and Rao, 1997] has the strongest similarities to the work presented in this paper. However, Kishore defined fuzzy sets to represent the step-size of the learning rate for training RBF networks.

7 Conclusions

Knowledge synthesis i.e. the modification of existing RBF networks using heuristic rules has obvious benefits when used in certain situations. The use of knowledge synthesis only makes sense when the available data is insufficient to build a reliable classifier. In such a situation it is advantageous to use heuristic rules to modify an existing RBF network to detect infrequently encountered input vectors that would otherwise be misclassified. However, care must be taken when applying the domain rules. Unless the domain rules can cover a large proportion of the synthesized input to output space it is likely that not all the necessary centre positions will be moved into appropriate locations. This effect will reduce the effectiveness of the new centres. Fortunately, because of the local characteristics of the RBF network the existing centres will be relatively unaffected and should still be able to classify with the same accuracy before knowledge synthesis occurred.

References

[Abu-Mostafa, 1993] Y. Abu-Mostafa. Hints and the VC dimension. *Neural Computation 5*, pages 278–288, 1993.

[Andrews and Geva, 1996] R. Andrews and S. Geva. Rules and local function networks. In *Rules and Networks-Proceedings of the Rule Extraction From Trained Artificial Neural Networks Workshop, Artificial Intelligence and Simulation of Behaviour*, Brighton UK, 1996.

[Casalino *et al.*, 1998] F. Casalino, F. Masulli, and A. Sperduti. Rule specialization in networks of fuzzy basis functions. *Intelligent Automation and Soft Computing*, 4(1):73–82, 1998.

[Halgamuge, 1997] S. Halgamuge. Self-evolving neural networks for rule-based data processing. *IEEE Transactions on Signal Processing*, 45(11):2766–2773, 1997.

[Hunt *et al.*, 1996] K. J. Hunt, R. Haas, and R. Murray-Smith. Extending the functional equivalence of radial basis function networks and fuzzy inference systems. *IEEE Transactions on Neural Networks*, 7(3):776–781, May 1996.

[Jang and Sun, 1993] J. S. R. Jang and C. T. Sun. Functional equivalence between radial basis function networks and fuzzy inference systems. *IEEE Transactions Neural Networks*, 4(1):156–159, January 1993.

[Jin and Sendhoff, 1999] Y. Jin and B. Sendhoff. Knowledge incorporation into neural networks from fuzzy rules. *Neural Processing Letters*, 10:231–242, 1999.

[Kishore and Rao, 1997] A. V. Kishore and M. V. Rao. A novel based algorithm for radial basis function network. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 489, pages 2007–2011, 1997.

[Kubat, 1998] M. Kubat. Decision trees can initialize radial basis function networks. *IEEE Transactions on Neural Networks*, 9(5):813–821, 1998.

[Mamdani and Baakini, 1974] E. H. Mamdani and Baakini. Prescriptive method for deriving control policy in a fuzzy-logic controller. *Electronics Letters*, 11(25/26):625–626, 1974.

[McGarry *et al.*, 2001] K. McGarry, S. Wermter, and J. MacIntyre. Knowledge extraction from local function networks. In *Seventeenth International Joint Conference on Artificial Intelligence*, volume 2, pages 765–770, Seattle, USA, August 4th-10th 2001.

[Mitra and Hayashi, 2000] S. Mitra and Y. Hayashi. Neuro-fuzzy rule generation: survey in soft computing framework. *IEEE Transactions on Neural Networks*, 11(3):748–768, 2000.

[Moody and Darken, 1989] J. Moody and C. J. Darken. Fast learning in networks of locally tuned processing units. *Neural Computation*, pages 281–294, 1989.

[Narazaki, 1996] H. Narazaki. Reorganizing knowledge in neural networks: An explanatory mechanism for neural networks in data classification problems. *IEEE Transactions on Systems, Man and Cybernetics*, 26(1):107–117, 1996.

[Roscheisen *et al.*, 1994] M. Roscheisen, R. Hofmann, and V. Tresp. Incorporating prior knowledge into networks of locally-tuned units. In *Computational learning Theory and Natural Learning Systems*, volume III, pages 53–64. 1994.

[Sun, 1994] C. T. Sun. Rule-base structure identification in an adaptive-network-based fuzzy inference system. *IEEE Transactions on Fuzzy Systems*, 2(1):64–73, 1994.

[Tresp *et al.*, 1997] V. Tresp, J. Hollatz, and S. Ahmad. Representing probabilistic rules with networks of gaussian basis functions. *Machine Learning*, 27:173–200, 1997.