Pergamon

Control Eng. Practice, Vol. 5, No. 5, pp. 683-692, 1997

0967-0661/97 $17.00 + 0.00

PII:S0967-0661(97)00050-6

METHODS OF FAULT DIAGNOSIS

S. Leonhardt and M. Ayoubi

Institute of Automatic Control, Darmstadt University of Technology, Landgraf-Georg-Str.4, 64283 Darmstadt, Germany
(sleonhardt@irt.th-darmstadt.de)

(Received December 1996; in final form March 1997)

Abstract : This paper gives a summary of methods that can be applied to automatic fault diagnosis. In the beginning, the focus is on classification and diagnostic reasoning using fuzzy logic. Subsequently, some of the ideas which have led to the emerging neuro-fuzzy algorithms are discussed. Finally, a new neuro-fuzzy algorithm which has recently been developed, is briefly described. *Copyright © 1997 Elsevier Science Ltd*

Keywords: Pattern recognition, Bayes classifier, nearest neighbour, polynomial classification, neural networks, fuzzy logic, neuro-fuzzy systems

1. INTRODUCTION

For several years, automatic diagnosis has been an active area of research in both technical and medical applications, see (Patton, et al., 1989; Schill, 1990; Isermann, 1994; Frank, 1994; Isermann, 1996) and it is fair to say that the field is now beginning to mature. This paper aims at presenting an overview of state-of-the-art techniques.

Generally, automatic fault diagnosis can be viewed as a sequential process involving two steps: the *symptom extraction* and the actual *diagnostic* task, see Fig. 1.

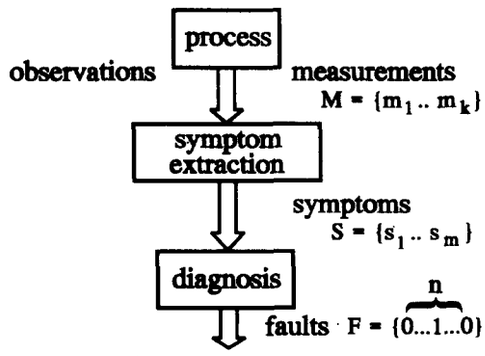

Fig. 1. Diagnostic concept

Symptom extraction is mainly required for data reduction and contrast amplification purposes. Many methods have been developed during the past few years, and the choice of a specific method is somewhat dependent on the nature of the process. Two general types can be distinguished :

- *analytic* symptom generation
- *heuristic* symptom generation

While analytic symptom generation is based on measurements, heuristic symptom extraction requires a human operator observing the process. Hence, analytic symptom generation is a continuous mapping from signal space to symptom space

$$M \in \mathbb{R}^k \rightarrow S \in \mathbb{R}^m , \tag{1}$$

while heuristic symptoms can be interpreted as a mapping of observations onto symptoms.

The focus of this paper will be the actual diagnostic algorithm. This second step is different from the first in that the output vector F is often assumed to be binary. Then, diagnosis is not a continous, but a discrete mapping.

$$S \in \mathbb{R}^m \rightarrow F \in \{0,1\}^n \tag{2}$$

Regarding their form of knowledge acquisition and interpretation, three types of algorithms can be distinguished :

- *classification* methods
- *inference* methods
- *combinations* like adaptive fuzzy-neuro systems.

Classification methods include geometric, statistic, neural and fuzzy classifiers and generally use **reference patterns** for learning. Since these algorithms are often applied to recognition problems, they are also referred to as *pattern- recognition* algorithms.

By contrast, *inference* methods are based on **linguistic rules** rather than reference patterns and are thus often called *rule-based* methods. The rules may be formulated in a crisp manner ("IF THEN ELSE - rules), but most of the time they are given in a fuzzy way. Expert systems fed with fuzzy rules allow a fuzzy decision making, the so called "approximate reasoning". Fig. 2 compares the two principles.

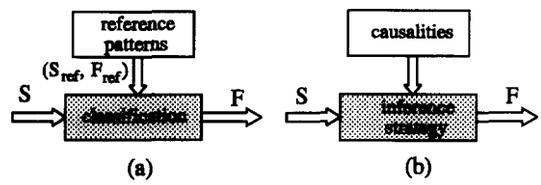

Fig. 2. Classification (a) and inference (b) methods

Due to the nature of the knowledge representation, classification methods are also referred to as *implicit* methods while the inference methods should be named *explicit* algorithms as the process knowledge is formulated in a more direct linguistic manner. Both approaches will be discussed in detail in Sections 2 and 3, respectively.

In order to combine the advantages of both approaches, recent research has focused on fusion of the two methods, see e.g. (Isermann and Ayoubi, 1996). Some combinations, referred to as *adaptive neuro-fuzzy* systems, will be presented in Section 4.

2. CLASSIFICATION METHODS

Throughout this section, it is assumed that there are p reference vectors (S_{ref},F_{ref}) available with $S \in \mathbb{R}^m$. It is further assumed that there are n faults F_i which are binary coded.

2.1. Statistical classifiers

These classification algorithms are based on certain assumptions about the symptom statistics. A well-known example is the Bayes classifier.

Suppose that the *conditional probability* $p(S|F_i)$, the *posterior probability* $p(F_i|S)$ and the *prior probability* $p(F_i)$ are known. Remember that $p(S|F_i)$ specifies the probability for the symptom vector S, given that it belongs to fault F_i, while $p(F_i|S)$ gives the probability of a certain fault once a specific symptom vector S has been observed. The prior probability $p(F_i)$ simply gives the probability of the fault F_i to occur. Then Bayes' theorem is given by

$$p(F_i|S) = \frac{p(S|F_i) \cdot p(F_i)}{p(S)} \tag{3}$$

where $p(S)$ gives the probability for a specific symptom vector S to occur. The risk of misclassification is minimized if, for a specific symptom vector S, the fault F_i with the highest posterior probability is chosen. Introduce a sym-metric cost function C defined by

$$C(F,\hat{F}) = \begin{cases} 0 \;\forall\; \hat{F}=F & (\textit{correct decision}) \\ C_w \;\forall\; \hat{F}\neq F & (\textit{wrong decision}) \\ C_r \;\forall\; \hat{F}=F_0 & (\textit{refuse to decide}) \end{cases}, \tag{4}$$

where $\hat{F}$ is the optimal estimate for F_i and F_0 stands for "don't know". Then the optimal decision rule is given by

$$\textit{find } \hat{F} \textit{ such that } \begin{cases} p(\hat{F}|S) = \max\{p(\hat{F}|S)\} \\ p(\hat{F}|S) > \dfrac{C_r - C_w}{C_r} \end{cases} \tag{5}$$

$$\textit{otherwise } \hat{F} \doteq F_0 \quad (\textit{refusal}) .$$

Note that this decison rule actually picks the fault class with the highest probability, but only if the probability for that specific class is higher than the relative risk of a wrong decision. By applying Bayes' theorem, the upper two lines of eq. (5) can be re-written as

$$\begin{aligned} p(S|\hat{F}) \cdot p(\hat{F}) &= \max_i \{p(S|F_i) \cdot p(F_i)\} \\ p(S|\hat{F}) \cdot p(\hat{F}) &> \frac{C_r - C_w}{C_r} \cdot p(S) . \end{aligned} \tag{6}$$

The importance of Bayes' theorem lies in the fact that it re-expresses the posterior probability in terms of quantities which are often much easier to calculate. For the conditional probability, assume a normal distribution with mean value μ and covariance matrix K. Assume further that the probability for all faults is $p(F_i) = p_F$. Thus, from eq. (6) a decision boundary line d_F

$$\begin{aligned} d_F(S) &= p(S|F_i) \cdot p(F_i) \\ &= \frac{p_F}{\sqrt{(2\pi)^m \cdot \det(K)}} \cdot e^{-\frac{1}{2}(S-\mu)^T \cdot K^{-1} \cdot (S-\mu)} \end{aligned} \tag{7}$$

can be computed. Since this decision is not influenced by a monotonic mapping, it is custom to scale eq. (7) by $(2\pi)^{m/2}$, and to take the ln leading to

$$d_F^*(S) = \ln(d_F(S)) = \ln(p_F) - \frac{1}{2}\ln[\det(K)] - \frac{1}{2}[(S-\mu)^T \cdot K^{-1} \cdot (S-\mu)] . \quad (8)$$

Note that these decision rules are quadratric functions. Thus, the resulting boundary lines which separate the areas belonging to a certain fault are actually conic sections. For m = 2 and n = 3, Fig. 3 shows an example.

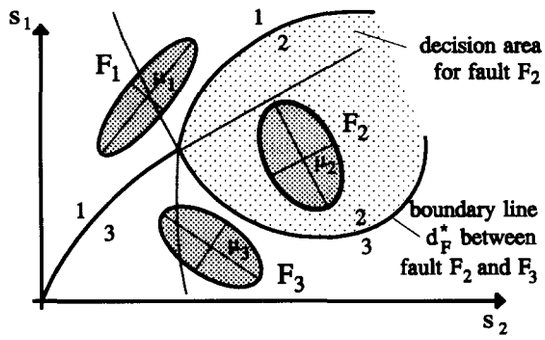

Fig. 3. Decision boundaries for the Bayes classifier

It should be pointed out that in order to use the Bayes classifier, the statistical properties have to be known. Although there are ways to estimate these statistics from data, this prerequisite has to some extent limited the practical applicability of Bayes classification. On the other hand, if these properties are known or satisfactorily estimated, the Bayes classifier is the optimal classifier.

2.2 Geometric classifiers

These classifiers belong to a simple yet quite attractive group of classification algorithms which are based on geometric distance computations, see (Dasarthy, 1990). The basic idea is that a certain symptom vector S_{test} should belong to the fault class of the reference vector to which it has the shortest distance. One possibility is to use the geometric distance Δ_i as defined by

$$\Delta_i = \sqrt{\sum_{k=1}^{m} (s_{test,k} - s_{ref,i,k})^2} . \quad (9)$$

Thus, the task is to find

$$\{i \in (1..p) \mid \exists (\min(\Delta_i))\} . \quad (10)$$

If i is determined, then the nearest neighbouring reference vector has been identified and the best guess for the fault class of the test vector S_{test} is F_i. Fig. 4 illustrates a possible scenario for the case m = 3 and n = 3.

The method can be generalized to include not just the closest, but the k nearest reference vectors. In this

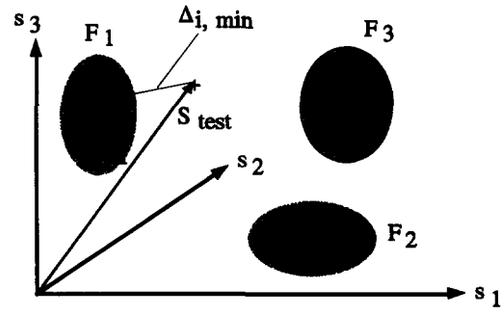

Fig. 4. Visualisation of geometric classification

case, the method is referred to as "k-nearest neighbour" algorithm. It may be necessary to determine the fault class even if more than one class is found among the k nearest reference vectors.

2.3. Polynomial classifiers

In Section 2.1, it has been shown that in case of a normal distribution, the Bayes classification algorithm results in quadratric decision rules leading to conic section boundary lines.

By contrast, polynomial classifiers assume that the decision rules may be sufficiently approximated by polynomials of order g, i.e. by

$$d_{F,i}(S) = w_{0,0} + w_{1,1} \cdot s_1 + \ldots + w_{1,g} \cdot s_1^g + w_{2,1} \cdot s_2 + \ldots + w_{2,g} \cdot s_2^g + \ldots + w_{m,1} \cdot s_m + \ldots + w_{m,g} \cdot s_m^g + w_{m+1,1} \cdot s_1 \cdot s_2 + \ldots \quad (11)$$

with $i \in \{1..n\}$. Let the vector $X(S)$ contain all polynomials and linear combinations of the symptom components. Then

$$d_{F,i}(S) = w_i^T \cdot X(S) . \quad (12)$$

If all n fault classes are considered, eq. (12) may be expanded to

$$d_F(S) = W \cdot X(S) . \quad (13)$$

Let $r = \dim(X(S))$. If all possible linear combinations of polynomials are considered, then

$$r = dim(X(S)) = \binom{m+g}{g} \quad (14)$$

and

$$dim(W) = n \cdot r . \quad (15)$$

Polynomial classifiers "learn" by finding an optimal estimate for the matrix W which yields

$$Q(W) = E\left\{\sum_{i=1}^{p} |F_{ref,i} - W \cdot X(S_{ref,i})|^2\right\} \stackrel{!}{=} \min. \quad (16)$$

It can be shown that eq. (16) is generally solved by

$$W_{opt}^T = E\{X(S)\cdot X(S)^T\}^{-1}\cdot E\{X(S)\cdot F\} \ . \quad (17)$$

Considering the p reference vectors (S_{ref}, F_{ref}), this leads to

$$W_{opt}^T = \left[\sum_{i=1}^{p} X(S_{ref,i})\cdot X(S_{ref,i})^T\right]^{-1}\cdot\left[\sum_{i=1}^{p} X(S_{ref,i})\cdot F_{ref,i}\right] . \quad (18)$$

Note that from the approximation theorem of Weierstraß, it is known that a continuous function may always be approximated by a polynomial of sufficiently high order to any degree of accuracy.

However, depending on the chosen order of the polynomials, the vector dimensions can become rather large, a major disadvantage of the polynomial classification approach.

2.4. Neural networks

Artificial neural networks (ANNs) are networks of simple, usually nonlinear, processing nodes which are "inspired" by the information processing in biological nervous systems. Very often, the processing nodes are Mc-Culloch-Pitts neurons (Fig. 5)

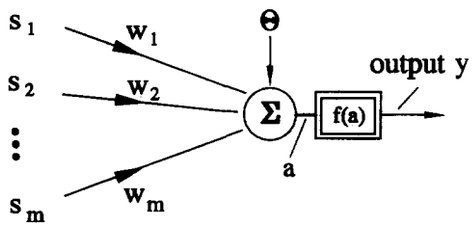

Fig. 5. McCulloch-Pitts neuron

These processing nodes imitate the information processing in the nervous system by computing the scalar product of an input vector S and a synaptic weight vector W plus an offset θ, leading to

$$a = \sum_{j=1}^{m} w_j \cdot s_j + \Theta \ . \quad (19)$$

Subsequently, the activation status a is mapped to the output via an nonlinear activation function f_{akt}. In the case of a sig-moidal function, this leads to

$$f_{sig}(a) = \frac{1}{1+e^{-a}} \ . \quad (20)$$

Within the large group of network architectures, the *multilayer perceptron* (MLP) is the best known example, see (Rumelhart and McCleland, 1986) and (Hecht-Nielsen, 1990). An MLP is composed of several layers each of which has a certain number of processing nodes. As an example, consider the case of three layers, see Fig. 6.

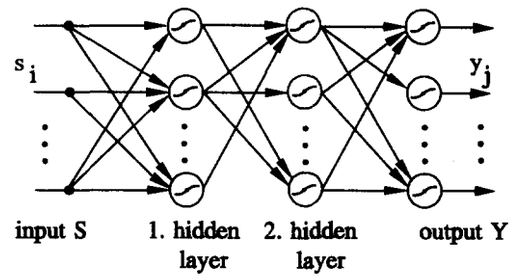

Fig. 6. Three-layer perceptron

In vector notation, the overall topology of this MLP is given by

$$Y = F_{sig,3}\left(\ldots\cdot F_{sig,1}\left(W^{[1]}\cdot X+\Theta^{[1]}\right)\ldots+\Theta^{[3]}\right) \quad (21)$$

where the activation vector F_{sig} of the j-th layer has dimension j^* and is defined by

$$F_{sig,j}^T(a) = \left[f_{sig}(a_0), \ \ldots, \ f_{sig}(a_{j^*-1})\right] . \quad (22)$$

The coupling between layers is performed by the weight matrix

$$W^{[j]} = \begin{bmatrix} w_{1,1} & \cdots & w_{1,j^*-1} \\ \vdots & & \vdots \\ w_{j^*,1} & \cdots & w_{j^*,j^*-1} \end{bmatrix} \quad (23)$$

in which each entry $w_{i',i}^{[j]}$ connects the i' th neuron of the j th layer with the i th neuron of the $(j\text{-}1)$st layer.

It has been shown by Funahashi (1989) and Hornik et al. (1989) that MLPs are also capable of approximating an arbitrary continuous function. Successful applications of MLPs to classification tasks have been frequently reported, see e.g. (Lippmann, 1987; Barschdorff, 1990; Bishop, 1995). MLPs "learn" by adjusting the different weight matrices, such that

$$Q\left(W^{[j]}\right) = \left\{\sum_{i=1}^{p} \left|F_{ref,i} - Y(S_{ref,i})\right|^2\right\} \stackrel{!}{=} \min. \quad (24)$$

For this task, there is a famous but sensitive and slow learning algorithm called "backpropagation", see (Rumelhart and McCleland, 1986). Due to the nested nonlinearites, it solves eq. (24) iteratively, and there is no guarantee of convergence. Thus, the user is forced to a "trial-and-error" approach which is known to be one of the major disadvantages of MLP networks.

From the fact that MLPs and polynomial classifiers are universal approximators, it must be concluded that - at least in the case of a single-layer perceptron - the polynomial classifier actually performs a Taylor expansion of the various sigmoids.

3. INFERENCE METHODS

3.1 Classical expert systems

Classically, the so-called *diagnostic reasoning* approach is performed by means of a rule base in which the single logical operations are stated as IF-THEN-ELSE rules like :

IF $<e_1$ AND $e_2>$ THEN $<F_1>$
IF $<s_1$ AND $s_2>$ THEN $<e_1>$
IF $<s_3$ OR $s_4>$ THEN $<e_2>$

The fact that the mapping of symptoms to faults is formulated in a verbal manner is essentially one of the major differences between inference methods and classification methods. Also, the inference methods allow one to obtain internal logical variables called "events" which already contain valuable information about the state of the process under observation. For the simple example given above, Fig. 7 shows a logical block diagram with four symptoms, two internal events and one fault.

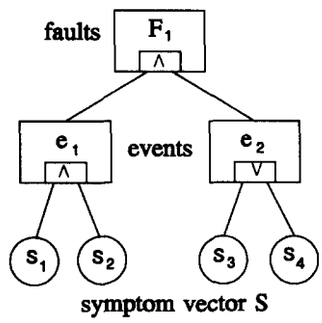

Fig. 7. Classical diagnostic reasoning (m = 4)

The major disadvantage of this approach has always been that binary logical decisions with Boolean operators do not reflect the gradual nature of many real-world diagnostic problems.

3.2 Fuzzy logic

The problem of vague information in diagnostic reasoning has been solved by Zadeh (1973), who introduced the concept of fuzzy logic. A certain symptom s_i is now evaluated according to its degree of membership of a certain specified "membership" function $\mu(s_i)$. The values of these functions are usually limited to 0 .. 1. Fig. 8. gives some examples.

The logical operations AND, OR and NOT are now applied to the membership functions instead of the symptom values directly. There are at least two different concepts to express these t-norms and t-co-norms, namely

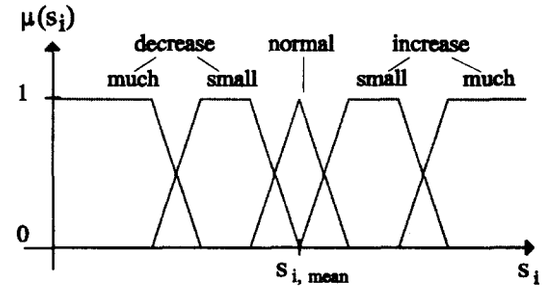

Fig. 8. Examples of membership functions

- the max-min notation :

$$\begin{aligned} &\textit{fuzzy AND}: \ \mu(e_i) = \min\left[\mu_1(s_1), \ldots, \mu_j(s_m)\right] \\ &\textit{fuzzy OR}: \ \mu(e_i) = \max\left[\mu_1(s_1), \ldots, \mu_j(s_m)\right] \\ &\textit{fuzzy NOT}: \ \mu(e_i) = 1 - \mu_j(s_m) \end{aligned} \tag{25}$$

- the prod-sum notation

$$\begin{aligned} &\textit{fuzzy AND}: \ \mu(e_i) = \prod_{j,k,m} \mu_1(s_1) \cdot \mu_2(s_j) \cdot \ldots \cdot \mu_k(s_m) \\ &\textit{fuzzy OR}: \ \mu(e_i) = 1 - \prod_{j,k,m} \mu_1(s_1) \cdot \mu_2(s_j) \cdot \ldots \cdot \mu_k(s_m) \\ &\textit{fuzzy NOT}: \ \mu(e_i) = 1 - \mu_k(s_m) \ . \end{aligned} \tag{26}$$

After the evaluation of the rule-base membership function, a defuzzification is required in order to obtain a "crisp" output, see (Driankov et al., 1993). Some of the known methods are illustrated in Fig. 9.

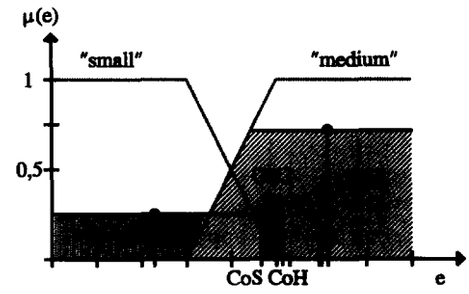

Fig. 9. Defuzzification methods

Note that CoG means "center of gravity" given by

$$out = \frac{\int e \cdot mu(e)\, de}{\int mu(e)\, de} \ . \tag{27}$$

Similiarily, CoM stands for "center of maxima", CoS denotes "center of sums" and CoH stands for "height defuzzification". Hence, when using fuzzy logic in a diagnostic environment, one has to perform the following successive steps :

- *fuzzification* of "crisp" values (measurements, symptoms) as illustrated in Fig. 7.
- *inference* using a rule base in which the logical

operations are performed on the membership functions, using either eq. (25) or eq. (26) and *aggregation*.
- *defuzzification* to obtain crisp outputs, using one of the methods shown in Fig. 9.

In order to stress that vagueness can be dealt with, the whole approach to diagnosis has been named *approximate reasoning*. Fig. 10 shows an example.

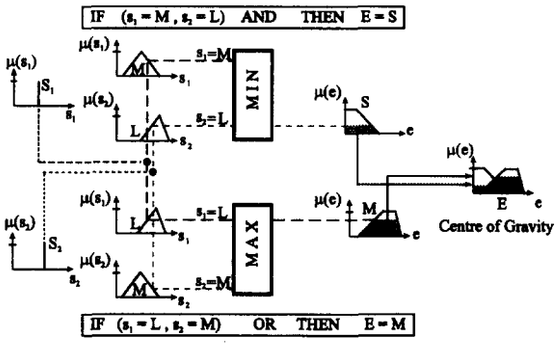

Fig. 10. Approximate reasoning for two inputs and two rules

4. ADAPTIVE NEURO-FUZZY SYSTEMS

The primary bottleneck of the *inference* systems is the tedious rule-base acquisition. It is a time-consuming task to observe the causalities between heuristic (or analytic) symptoms and the corresponding faults, and to formulate proper rules. In addition, the necessity of continuously tuning the knowledge-base in order to acquire consistent rules requires an adaptive component which adjusts the rule-base to match the problem context under consideration. With respect to the numeric knowledge representation, the *classification* algorithms represent a complementary approach to rule-based systems. However, their structure is not very transparent, as they approximate an arbitrary black-box model of the mapping rule. Furthermore, a priori expert knowledge cannot be considered, which would be a good way to initialize the parameters in some of the network-oriented classifiers and improve learning convergence.

The prime concern in the fusion of neural networks and fuzzy systems is to combine the strengths of both theories in order to achieve adaptive (learning) diagnostic systems with a transparent knowledge representation. Several concepts are possible, see e.g. (Preuß and Tresp, 1994; Nauck et al., 1994; Bonfig, 1995; Ayoubi and Isermann, 1997). These can essentially be classified in four major categories.

4.1 Neural networks influenced by fuzzy logic

One possible way to combine neural networks and fuzzy systems is to utilize expert knowledge presented as fuzzy IF-THEN rules in order to design and train neural networks. Clearly, such approaches focus mainly on the improvement of neural networks.

4.1.1 Fuzzy models within neural networks. In this sub-category, the basic idea is extend the McCulloch-Pitts neuron in order to "cope with fuzzy information", see Fig. 11. The fuzzy neuron is designed to function in much the same way as a crisp neuron does, except that it reflects the fuzzy nature of a neuron and has the capability to cope with fuzzy information. The inputs of such a fuzzy neuron can have crisp data or fuzzy sets ΔS_i in the universes of a given discourse. Depending on which neuron model is proposed, the synaptic operation differs from the determination of a membership degree to the evaluation of a fuzzy relation among the fuzzy input sets.

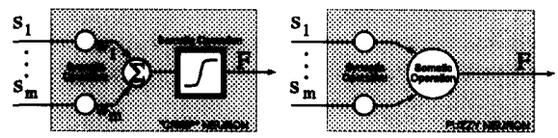

Fig. 11. (a) crisp neuron (b) fuzzy neuron

One possible fuzzy neuron model was described by logical equations in (Kiszka and Gupta, 1990). Here, the fuzzy neuron inputs are related to the outputs by a fuzzy conditional statement or by an IF-THEN rule as

$$R = s_1 \; x \; \ldots \; x \; s_m \; x \; F \; . \qquad (28)$$

For a given relation, the neuron output is calculated as :

$$F = s_1 \; o \; \left(\ldots \left(s_m \; o \; R \right) \ldots \right) \; . \qquad (29)$$

Another possibility is to treat the neuron inputs as crisp values. The *synaptic* operation then considers the determination of the membership degrees of each input to a given subset $\mu(s_i)$. The *somatic* operation is usually a kind of a t-norm or t-conorm operation, and the neuron activity is determined by

$$\mu(s_1, \ldots, s_m) = \mu(s_1) \otimes \ldots \otimes \mu(s_m) \qquad (30)$$

where $\otimes$ denotes the aggregation operator (Gupta and Qi, 1992; Ishibushi et al., 1994; Miyazaki et al., 1994).

Neural networks which consist of fuzzy neurons are called *fuzzy-neural networks*. The adaptation of these networks is carried out by the modification of the membership functions. Yamakawa and Tomoda (1989) describe the application of such a fuzzy-neural network to a classification problem.

4.1.2 Fuzzy-based adaptation of neural networks. Here, the main objective is to "improve the adaptation and learning of neural networks". Fig. 12 illustrates the concept.

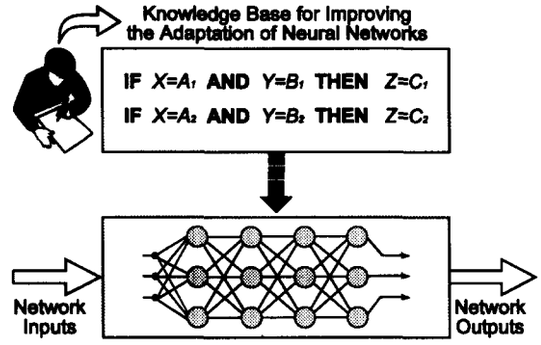

Fig. 12. Fuzzy-based adaptation of neural networks

This can be carried out by designing a rule-based knowledge frame by experts who are frequently involved with the training of neural networks. For example, Lin and Cunningham (1994) describe the application of the so-called *fuzzy delta-rule* which presents an extension of the known error back-propagation algorithm used to train MLPs.

These authors also use fuzzy logic for setting the initial weights of the neural network, in order to shorten the adaptation times and improve the convergence of the training procedure. Various other approaches to map fuzzy inference mechanisms to a neural network have been proposed, see (Berenji, 1992; Brown and Harris, 1994; Halgamuge and Glesner, 1994; Kruse et al., 1994).

4.2 Fuzzy systems influenced by neural networks

The second kind of integration approaches focuses on the *extension of fuzzy systems by means of ANNs.* Basically, two techniques can be distinguished :

- the neural network and the fuzzy system are connected in a *parallel* or *serial* configuration (no topological equivalence between the two techniques is required)
- the neural network is *topologically* designed to copy the fuzzy system.

4.2.1 Parallel/serial configuration. This approach to integrating fuzzy systems and neural networks considers a *block-oriented* combination of both methodologies. The ANNs are supporting the fuzzy system in a parallel or a serial manner. The primary feature of such approaches is that the two sys-tem types do not necessarily relate to each other with respect to their topology.

The *parallel configuration* can either be used to extend the resulting output of the fuzzy system additively, by means of the neural network, or to apply neural optimization procedures in order to investigate and tune the parameters of the fuzzy system. Application examples for this approach can be found in (Wöhlke, 1994) and (Perneel et al., 1994).

The *serial configuration* addresses applications where the neural network preprocesses the input data of the fuzzy system in order to perform data reduction, clustering and noise suppression. For instance, Nie et al. (1994) and Ultsch and Korus (1995) describe the application of self-organizing neural networks to initialize the rule-base of the fuzzy system. Both concepts are illustrated in Fig. 13.

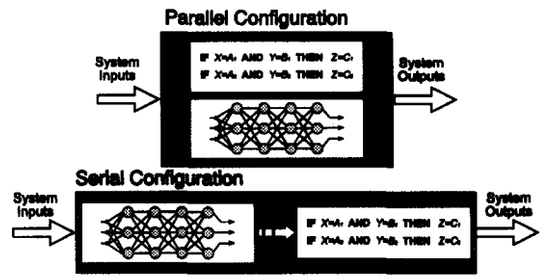

Fig. 13. ANN-based supplement of fuzzy systems in a block-oriented configuration

4.2.2 Neural network-based adaptation of fuzzy systems. In recent years, extensive research has been done to evolve special neural-network models whose topology is equivalent to the structure of IF-THEN rules in fuzzy systems. These so-called *neuro-fuzzy systems* possess the ability to emulate fuzzy inference mechanisms. The prime concern there was to combine the strengths of both theories in order to achieve adaptive (learning) systems with transparent knowledge representation featuring :

- transparent structures allowing extracted knowledge from the ANN to be automatically adopted to the rule-base.
- a direct relationship between the weights of the neuro-fuzzy system and the parameters of the rule base (e.g., rule priorities, relevance of inputs, etc.).
- existing expert knowledge can be utilized to initialize the neuro-fuzzy network in order to achieve better and faster convergence of the training procedure.
- extracted rules within the neuro-fuzzy systems can be verified in a straightforward way by the human experts for plausibility and interpretation.

Thus, the primary desire is to extend knowledge-based expert systems by an adaptive component that realizes a learning-like capability. The learning component has to facilitate the bidirectional knowledge exchange between the neural network and the fuzzy

system, in order to enable a permanent adaptation and tuning of the rule base with less effort, see Fig. 14.

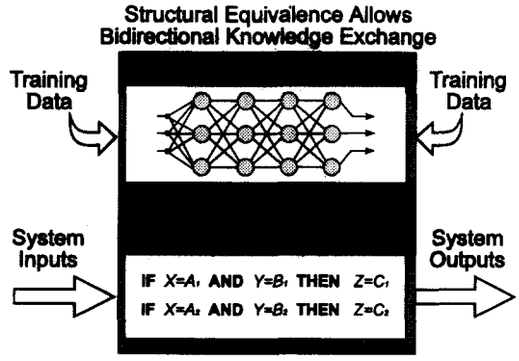

Fig. 14. Neural-network-based adaptation of fuzzy systems

It should be mentioned that neuro-fuzzy systems are nonlinear approximators, and identify a decision surface in an input/output sense. Thus, it cannot generally be expected that the network will also identify the internal structure of the underlying rule-base.

In fact, a neuro-fuzzy network can identify the desired rules only if the structure of the rules corresponds to the structure of the network. Still, the extracted rules remain transparent and can be verified by the expert for plausibility.

4.3 Hybrid fuzzy-neuro systems

Note that each circle denoting a neuron stands for a different operation within a fuzzy framework:

- neurons with a radial basis function (RBF) model the subsets of the linguistic variables
- neurons with a sigmoidal activation function which identify the logical premise operation

In hybrid fuzzy-neuro systems, the integration and interaction of the rule-based parts and the neural component is so strong that it becomes difficult to identify the origin and to separate the different fractions of a specific approach.

Ayoubi (1996) recently presented an example of such a system named SARAH (a *System for Adaptive Rule Acquisition with Hebbian Learning*), which includes two neuron types: *radial basis functions* and *perceptron* neurons, see Fig. 15.

The whole network consists of three layers according to the three inference steps of fuzzy approximate reasoning.

- *antecedent layer*: includes one-dimensional radial basis functions to model fuzzy subsets of the inputs (e.g. small, large, etc.). This layer performs the fuzzification of the crisp network inputs.

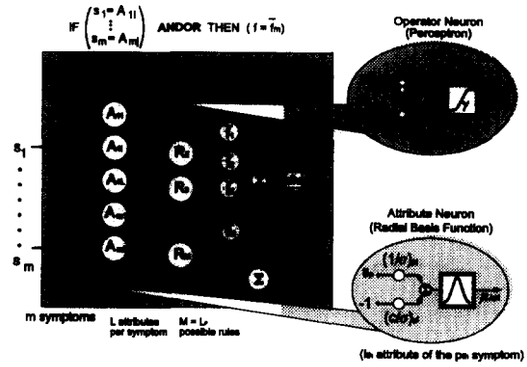

Fig. 15. Neuro-fuzzy network with m inputs and one output

The output of the layer expresses the degree of membership of each input to the implemented fuzzy subsets. The parameters of the Gaussian neurons are estimated from data using a fuzzy clustering algorithm.

- *relation layer*: consists of perceptron neurons that realize the logical conjunction or disjunction of the antecedents from the first layer, depending on the sigmoidal activity function threshold α_0 of each perceptron neuron. This layer evaluates the composition of the antecedents in order to investigate the grade of fulfilment for each rule. The input weights of the sigmoidal neurons consider the relevance of the rule premises to the rule conclusion.
- *conclusion layer*: includes the conclusions of the rule-base presented by the singletons. This layer aggregates the rule-base of the second layer and calculates the crisp output.

In order to illustrate the operation of the network, the following example with a two-rule base is presented:

$$\text{IF } (s_1 = \textit{medium}) \text{ AND } (s_2 = \textit{large}) \text{ THEN } (F = \textit{small})$$
$$\text{IF } (s_1 = \textit{large}) \text{ OR } (s_2 = \textit{medium}) \text{ THEN } (F = \textit{medium})$$

Each one-dimensional Gaussian basis function determined by its centre c and variance σ^2 models a fuzzy subset of the inputs (e.g. "large"). Within the second layer, the sigmoidal neurons realize the logical conjunction or disjunction, depending on the threshold of the sigmoidal activity function. The input weights consider the relevance of the rule premise (s_1 *is large*) to the conclusion of the rule. The conclusions are presented by singletons, see Fig. 16.

Note that the neurons with Gaussian functions actually model the fuzzy subsets while, depending on the threshold α, the sigmoid neurons model the logical operators AND or OR, respectively. An application of the concept to fault diagnosis including automatic rule extraction has been presented in (Ayoubi, 1996).

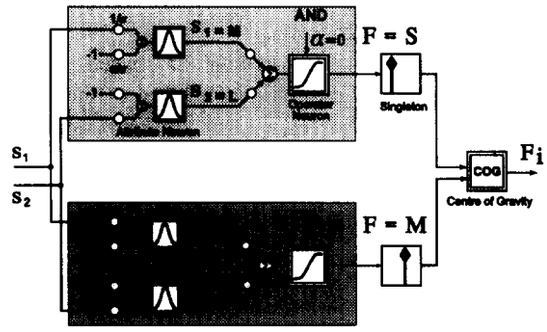

Fig. 16. The inference in the neuro-fuzzy network

5. CONCLUSIONS

An overview of the different approaches to fault diagnosis has been given, with a special focus on recent developments in the area of fuzzy logic and neural networks. The principles behind the different methods have been sketched and, in some cases, the major disadvantages have been mentioned.

So far, none of the methods presented solves the remaining *problem of completeness*. Thus, in practical applications, unexpected symptom combinations may still cause problems, if they are not covered by a rule or data set.

6. REFERENCES

Ayoubi, M. and Isermann, R. (1997). Neuro-fuzzy--systems for diagnosis. *International Journal for Fuzzy Sets and Systems* (to be published).

Ayoubi, M. (1996). Fuzzy systems design based on a hybrid neural structure and application to the fault diagnosis of technical processes. *Control Engineering Practice*, Vol. 4, No 1, pp. 35-42.

Barschdorff, D. (1990). Case studies in adaptive fault diagnosis using neural networks. *IMACS/IFAC Int. Symp. on Math. and Int. Models in System Simulation.* Brussels, Belgium, pp. 411-416, Sep. 3 - 7.

Berenji, H. (1992). Learning and tuning fuzzy logic controllers through reinforcements. *IEEE Trans. on Neural Networks*, Vol. 3, No 5 (1992).

Bishop, C. (1995). *Neural networks for pattern recognition.* Clarendon Press, Oxford.

Bonfig, K.W. (1995). *Neuro-Fuzzy : Grundlagen und Anwendungen in der industriellen Automatisierung.* Expert-Verlag, Renningen.

Brown, M. and Harris, C. (1994). *Neurofuzzy adaptive modelling and control.* Prentice Hall, New York.

Dasarthy, B. (1990). *Nearest neighbour pattern classification techniques.* IEEE Computer Society Press, Los Alamitos, CA.

Driankov, D., Hellendorn, H. and Reinfrank, M. (1993). *An introduction to fuzzy control.* Springer Verlag, Berlin.

Frank, P.M. (1994). Diagnoseverfahren in der Automatisier-ungstechnik. *Automatisierungstechnik (at)*, Vol. 42, No. 2, pp. 464-477.

Funahashi, K. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks*, Vol. 2, pp. 183-192.

Gupta, M. and Qi, J. (1992). On fuzzy neuron models (in *Fuzzy logic for the management of uncertainty.* eds. Zadeh, L.A and Kacprzyk, J.). John Wiley & Sons, New York.

Halgamuge, S. and Glesner, M. (1994). Fuzzy neural fusion techniques for industrial applications. *ACM Symp. on Applied Computing*, Phoenix, AZ.

Hecht-Nielsen, R. (1990). *Neurocomputing.* Addison-Wesley Pub. Company, Reading, MA.

Hornik, K., Stinchcombe, M. and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, Vol. 2, pp. 359-366.

Ishibushi, H., Morioka, K. and Tanaka, H. (1994). A Fuzzy neural network with trapezoid fuzzy weights. *3rd IEEE Conf. on Fuzzy Systems.* Orlando, FL, pp. 228-233.

Isermann, R. (1994). *Überwachung und Fehlerdiagnose.* VDI Verlag, Düsseldorf.

Isermann, R. (1996). Modellgestützte Überwachung und Diagnose Technischer Systeme. *Automatisierungstechnische Praxis (atp)*, Vol. 38, No. 5+6.

Isermann, R. and Ayoubi, M. (1996). Fault detection and diagnosis with neuro-fuzzy-systems. *EUFIT '96.* Aachen, Germany, Sep. 2-5.

Kiszka, J. and Gupta, M. (1990). Fuzzy Logic Neural Network. *BUSEFAL 4*, pp. 104-109.

Kruse, R., Gebhardt, J. and Klawonn, F. (1994). *Foundations of fuzzy systems.* Wiley, Chichester.

Lin, Y. and Cunningham, G. (1994). A new fuzzy approach for setting the initial weights in a neural network. *3rd IEEE Conf. on Fuzzy Systems*, Orlando, pp. 40-45.

Lippmann, R.P. (1987). An introduction to computing with neural nets. *IEEE ASSP Magazine*, No. 4, pp. 4-22.

Miyazaki, A., K. Kwon, K., Ishibuchi, H. and H. Tanaka, H. (1994). Fuzzy regression analysis by fuzzy neural networks and its application. *3rd IEEE Conf. on Fuzzy Systems*, Orlando, FL, pp. 52-57.

Nauck, D., Klawonn, F. and Kruse, R., (1994). *Neuronale Netze und Fuzzy-Systeme.* Vieweg Verlag, Braunschweig.

Nie, J., Loh, A. and Hang, C. (1994). Fuzzy modelling of nonlinear pH processes through neural approach. *3rd IEEE Conf. on Fuzzy Systems.* Orlando, FL, pp. 1224-1229.

Patton, R., Clark, R.N. and Frank, P.M. (1989). *Fault diagnosis in dynamic systems.* Prentice Hall, New York.

Perneel, C., Renders, J., Themlin, J. and Acheroy, M. (1994). Fuzzy reasoning and neural networks for decision making problems in uncertain environment.

3rd IEEE Conf. on Fuzzy Systems. Orlando, FL, pp. 1111-1125.

Preuß, H.-P. and Tresp, V. (1994). Neuro-Fuzzy. *Automatisierungstechnische Praxis (atp)*, Nr. 5, pp. 10-24.

Rumelhart, D.E. and McCleland, J.L. (1986). *Parallel distributed processing*. MIT Press, Cambridge.

Schill, K. (1990). *Medizinische Expertensysteme : Methoden und Techniken*. Oldenburg Verlag, München.

Ultsch, A. and Korus, D. (1995). Erwerb von Fuzzy-Wissen aus selbstorganisierenden Neuronalen Netzen. *3rd GI-Workshop on Fuzzy-Neuro-Systems*. Darmstadt, pp. 325-332.

Wöhlke, G. (1994). A neuro-fuzzy-based system architecture for the intelligent control of multi-finger robot hand. *3rd IEEE Conf. on Fuzzy Systems*. Orlando, FL, pp. 64-69.

Yamakawa, T. and Tomoda, S. (1989). A fuzzy neuron and its application to pattern recognition. *3rd IFSA*, Seattle, Washington, pp. 30-38.

Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transaction on Systems, Man and Cybernetics*. Vol. 3, pp 28-44.