Comments and questions can be sent to: michael@cs.uu.nl

ELSEVIER

Artificial Intelligence in Medicine 6 (1994) 359–381

Artificial Intelligence in Medicine

On the quality of neural net classifiers

Michael Egmont-Petersen [a], Jan L. Talmon [a,*], Jytte Brender [b], Peter McNair [c]

[a] *Dept. of Medical Informatics, University of Limburg, PO Box 616, 6200 MD Maastricht, The Netherlands*
[b] *Medical Informatics Laboratory Aps, Stengaards Allé 33d, DK-2800 Lyngby, Denmark*
[c] *Dept. of Clinical Chemistry, University of Copenhagen, Hvidovre Hospital, Kettegaard Allé 30, DK-2650 Hvidovre, Denmark*

Received June 1993; revised February 1994

Abstract

This paper describes several concepts and metrics that may be used to assess various aspects of the quality of neural net classifiers. Each concept describes a property that may be taken into account by both designers and users of neural net classifiers when assessing their utility. Besides metrics for assessment of the correctness of classifiers we also introduce metrics that address certain aspects of the misclassifications. We show the applicability of the introduced quality concepts for selection among several neural net classifiers in the domain of thyroid disorders.

Keywords: Neural net classifiers; Quality assessment; Quality metrics; Thyroid disorders

1. Introduction

Neural nets (NNs) are being used for classification/diagnostic tasks in various domains, including medicine [8,18,20,21,23,27,35,36,41]. Various issues play a role during the design of such classifiers. Until now, there hardly have been strict rules for how to design the networks apart from trial and error. It is possible to generate several hundreds of networks based on various settings of design parameters. To find the 'optimal' network for a specific classification task some yardstick or metric is required, which facilitates a comparison of different NNs. This problem may be approached in two different ways. First, it is possible to measure the degree to

* Corresponding author. Email: talmon@mi.rulimburg.nl

SSDI 0933-3657(94)00013-I

which the network has generalized the information contained in the set of training instances. Second, the performance of the network can be measured for another set of data. This process is often called cross validation when the metric used reflects how well the correct class labels are assigned [17,40]. These measures provide only limited insight in the performance of a network.
An additional problem is that what is an optimal network will depend on the situation in which the network will be applied. The requirements regarding the network's performance will be different in a screening situation as compared to its application in a highly specialized clinic.

In this paper, we propose several *quality concepts* and *quality metrics* that facilitate *quality assessment* of NN classifiers. The metrics are used to derive actual *quality measures* for a specific NN and they provide the potential user with means to select the most appropriate network among different nets.

In the following, we will briefly describe the kind of NNs we have been experimenting with. We will discuss issues that play a role in the design of these networks. Next we will introduce a number of quality concepts. Each of these quality concepts describes a *property* of a (neural net) classifier. Most of these quality concepts are general and can be applied to other classifiers as well.
We will define a set of metrics that can be used to measure the extent to which specific *properties* are present or absent. We show their applicability for networks in the domain of thyroid disorders.

2. Neural-net classifiers

An NN classifier consists of a set of interconnected processors. The way the processors are connected and what processes are performed by the interconnected processors determine their properties (see for an overview e.g. [39]). We will restrict ourselves to feed-forward networks that are trained by means of the back-propagation learning mechanism [34]. Feed-forward networks consist of a series of (fully) interconnected layers of processing units called nodes or neurons – see Fig. 1.

The first layer – the input layer – takes as input the various attribute values. The output of the nodes in the input layer, multiplied with the weights attached to the links, is passed to the nodes in the hidden layer.

A hidden node collects the incoming weighted output values of the previous layer. Besides that, it receives also the weighted value of a bias node. This bias node always outputs the value 1. It allows for adding an offset to the sum of the weighted inputs, similar to an offset in a regression or discriminant function.
The sum of the weighted input values is passed through a nonlinear *activation function* (see Fig. 2). Various types of activation functions have been proposed, such as sigmoidal, hyperbolic-tangent or logistic functions. The only requirements are that the output values of the function are bounded to an interval and that the nonlinear function can be differentiated. To avoid saturation of the nonlinear functions during training, the total input activation has to be bounded. This can be

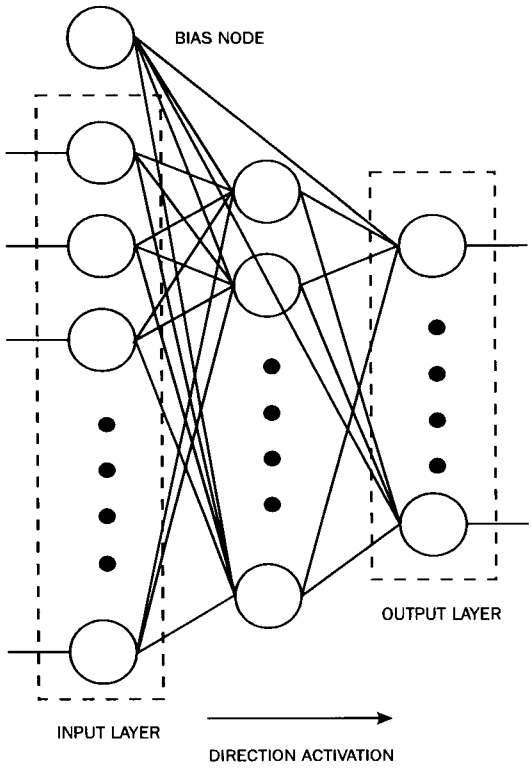

Fig. 1. General structure of a feed-forward network with one hidden layer.

achieved in two ways. One either scales the weights from each input node such that the orders of magnitude of the input value and the weights are reciprocal to each other. The other way is to scale the input values to a certain range. We selected a scaling for each input variable to the interval [0, 1]. By doing so, we can use initial weights of the same order of magnitude throughout the network.

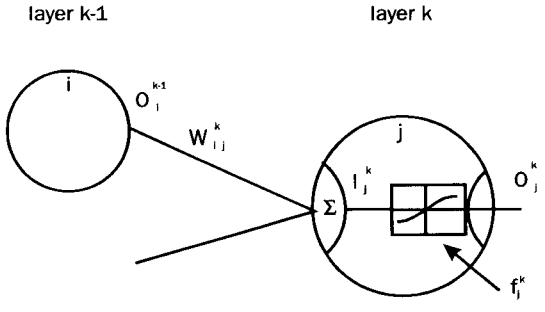

Fig. 2. The structure of a node in the hidden and the output layer.

The output of a node in the hidden layer is fed into the nodes of the next layer, again multiplied with the weights attached to the links. In principle, several hidden layers may be used. However, it has been shown that any arbitrary discriminant function can be built with a classifier with only one hidden layer, if enough hidden nodes are used [25,30]. The nodes of the output layer have the same structure as the nodes of the hidden layer(s).

What remains to be done, is to define how the required output – in our case class membership – is mapped on the output range of the function. When the limit values of the activation functions are used, the function has to be in saturation to achieve a good classification. This is often not possible. Therefore, several researchers have proposed to do a different mapping of the desired values, for example on $[-0.9, 0.9]$ or on $[0.1, 0.9]$. In our experiments we use the hyperbolic-tangent function and map the desired output on 0 (not belonging to the class) and 1 (belonging to the class).

The functioning of a node in a hidden layer or the output layer can be described as follows. The input to node j in layer k is given by

$$I_j^k = \sum_i O_i^{k-1} \times W_{ij}^k \tag{1}$$

where O_i^{k-1} is the output of node i in the previous layer, W_{ij}^k is the weight associated with the link between the nodes i and j and I_j^k is the input to the activation function f_j^k of node j in layer k. The summation over nodes i in (1) also includes the bias node.

The output of the node follows from

$$O_j^k = f_j^k\left(I_j^k\right) \tag{2}$$

To each node in the output layer a class label is assigned. The output value of such a node can be interpreted as the amount of evidence in favor of the corresponding class.

For an optimal network, the weights have to be given values such that for each case the appropriate output is generated. Such a configuration of weight values cannot be set a priori. This has led to the development of learning algorithms that adapt an initial weight configuration by successive passes through a set of learning cases. For each case, the proper class label is known, i.e. it is specified which output node should have as output the value corresponding to 'belonging to the class,' while all others should have as output the value corresponding to 'not belonging to the class'. We denote the required output at output node j for case p as $c_{j,p}$. The measure of the total error for all cases in the learning set, given a certain weight configuration, is given by

$$E = \tfrac{1}{2}\sum_p \sum_j \left(c_{j,p} - O_{j,p}^n\right)^2 \tag{3}$$

with $O_{j,p}^n$ being the output at node j in the output layer for case p.

By backpropagation of the errors in the network from the output layer through the hidden layer(s) to the input layer, the weights in the network are updated according to

$$\Delta_p w_{ij}^k = \eta \delta_{j,p}^k O_{j,p}^k \tag{4}$$

with η being the learning rate and $\delta_{j,p}^k$ dependent on whether one has to update the weights of links entering a node in the output layer or entering a node in one of the hidden layers. We refer the reader to ([34], pp 322–328) for the derivation of the formulas. Here, we will give only the results.
For links entering the nodes in the output layer $\delta_{j,p}^n$ is defined as

$$\delta_{j,p}^n = (c_{j,p} - O_{j,p}^n) f_j^{n'}(I_{j,p}^n) \tag{5}$$

in which $f_j^{n'}$ is the derivative of the activation function of node j in the output layer n. For links entering the nodes in the hidden layers, $\delta_{j,p}^k$ is defined as

$$\delta_{j,p}^k = f_j^{k'}(I_{j,p}^k) \sum_l \delta_{l,p}^{k+1} W_{jl}^{k+1} \tag{6}$$

This learning algorithm is repeatedly applied with the same learning set until some stop criterion is met (percentage of correctly classified cases greater than some threshold, total error, E, less than some threshold, numbers of passes through the learning algorithm greater than some threshold L, etc.). Rather than adapting the weights after processing of each case, the weights in the network were updated after each complete pass through all learning cases, with

$$\Delta w_{ij}^k = \sum_p \Delta_p w_{ij}^k \tag{7}$$

For the design of a feed-forward NN, the following issues play a role:

- *The number of hidden layers and the number of nodes for each hidden layer.* We will restrict ourselves to networks with one hidden layer. How to decide about the number of nodes in the hidden layer, is not straightforward [19,36,37]. It has been argued that the degrees of freedom in the network, defined as the number of weights – including those for the bias terms – has to be less than the number of cases in the learning set [2]. We will restrict ourselves in our experiments to networks that fulfil this requirement.
- *The learning rate η has to be selected.* The learning algorithm searches the space of weight values for an optimal solution, using a gradient descent approach. The learning rate defines the unit step size for ΔW. Therefore, like in all gradient methods, a small value for the learning rate will make the convergence of the network towards a solution slow, whereas a large value may cause the weights in the network to 'jump' back and forth over the appropriate value. We used a small learning rate, taking the slow convergence for granted.
- *The composition and size of the learning set.* As the network learns a mapping between input attribute values and class labels from a set of learning cases, the composition of the learning set plays a role in deriving networks that have

predictive value for new cases. In our experiments we will show that various learning sets, randomly drawn from a larger database, may result in networks with a significantly different performance.

- *The initial weights of the network.* The network has to start with some weight configuration before learning can begin. One often uses a random generator to provide the initial weights. It will be clear that the number of iterations required to achieve a well-performing network will depend on this weight configuration. Furthermore, it is known that the final weight configurations that result from different initial configurations need not to be the same, even when the networks are trained with the same learning set. During learning, the network in general gets stuck in a local minimum. So there is no guarantee that the absolute minimum of the total error is found. However, it has been reported that poor local minima are in practice rarely encountered [22]. To avoid these problems, an algorithm has recently been proposed that provides weight configurations that are closer to the optimal than some randomly generated configuration [29]. In our experiments, we used randomly generated initial weights to study their effect on the performance of the classifiers.
- *The type of nonlinear activation function.* As there is no theoretical argument why one function is better than the other, we have selected more or less arbitrarily a hyperbolic-tangent function in all our experiments.
- *The interpretation of the pattern of output values at the nodes in the output layer.* Most often a 'winner takes it all' approach is used. In this approach the class label belonging to the output node with the highest output is assigned to a case analyzed. This approach has some drawbacks. When all output levels are low, the case has apparently an attribute pattern that is not recognized by the classifier. It seems more useful to leave such a case unclassified. Also when two or more nodes have high output values, it is not clear which class label has to be assigned. A slight variation in the input attribute values may cause a completely different class assignment. For that reason we have selected an approach in which a class label is assigned to a case when one and only one output node has an output that exceeds a (node-dependent) threshold. (In our experiments we have set the threshold for all nodes at 0.5, the range of possible outputs being $[-1..1]$). When this requirement is not fulfilled, the case stays unclassified.

From the above discussion, it is clear that guidelines for how to handle these issues hardly exist. The developer of a neural classifier is left only with the option to generate a large number of neural classifiers using different initial settings and then to select the best one(s).

3. Quality concepts

In the domains of medical informatics, pattern recognition and statistics the quality or performance of classifiers is mostly addressed by means of some measure for the extent to which a classifier assigns the correct class labels [16]. This is, however, only one view on the quality of a classifier. An extensive

literature survey in various domains such as laboratory medicine, computer science and health-care management, revealed many quality concepts. Based on this review, preliminary definitions for these concepts were established [4–6]. From this large set, a number of essential quality concepts have been selected and a few new ones added that are relevant to describe the properties and applicability of NN classifiers. In this paper, we will deal only with a subset of these concepts as metrics have not yet been defined for all of them.

Before we embark on the definition of the quality concepts we need to consider what quality is. Brender defined quality in relation to the validation of information systems as:

"...the degree of fulfilment of the users' expectations" [3].

As the expectations may vary among users, this definition implies that quality cannot be expressed by one quantity. However, it may be represented by a set of quality characteristics (a quality profile).

A SYDPOL working group argued that

"Another aim [of quality assessment of decision support systems] is to emphasize the invisible properties of DSS, often concerned with the system's theoretical basis and prerequisites" [32].

We concluded that quality assessment of both information systems and decision support systems deals with the quantification of properties of the system under analysis. Such properties should assist the potential users in deciding whether the system is of any use in their situation.

An important prerequisite for a useful set of quality concepts is that each concept describes some distinct property of a classifier. This is necessary to characterize a specific classifier and to predict its behaviour for new cases.

Furthermore, the concepts should be generally applicable. When a concept is domain dependent, it will have hardly any value for other applications. Also the independence of the type of classifier analyzed is of value as it will facilitate a comparison of several types of classifiers by means of quality measurements.

It is obvious that the concepts should be interpretable by a (potential) end user of the classifier. When end users cannot interpret a quality concept and measurement, they cannot infer the classifier's appropriateness for their specific situation. Consequently, the users have to know the assumptions on which the metrics rely. For example, the users have to know how the *predictive value* of an outcome of a classifier depends on the prevalence of the classes and what information is necessary to recalculate the predictive value for their own situation.

Most quality concepts currently in use in the assessment of classifiers deal with *success* issues, i.e. they describe how well a classifier is doing on a certain problem. In some domains, like clinical chemistry, one also tries to quantify and to get insight in the *failure* characteristics of the evaluated processes. Examples are concepts like *precision* and *accuracy* for biochemical assays [9,31]. These concepts describe the variability in repeated measurement values from the same sample and how close these values are to the real value, respectively. Knowing what is wrong helps in deciding what to do to correct or improve the situation.

In the following we will briefly describe some of the concepts we developed and

used for the assessment of NN classifiers. More details and additional concepts can be found in [4–6].

3.1. Success quality concepts

There are two *success* quality concepts we will deal with in this paper, viz. *coverage* and *correctness*

Coverage is defined as:

> The extent to which a classifier is able to assign a class label to cases.

It may seem that this concept is introduced mainly because of our approach of interpreting the values of the output nodes of an NN. The concept is, however, equally applicable to other classifiers. An example is a linear discriminant function for which no decision is made when the function value is in a certain range. When the failure of a classifier to assign a class label is quantified, the concept *rejection rate* is often used [10,18].

Correctness is defined as:

> The extent to which a classifier is able to assign the correct class label to the cases that are covered by the classifier.

This is a commonly used concept in assessment of classifiers. Note that in our definition it only pertains to cases that receive a class label by the classifier.

When one is interested in the extent to which a classifier assigns proper class labels in the whole set of cases, a concept named *accordance* can be used. *Accordance* will only be used in our experiments during the learning of NNs as its metric is a composite of the metrics for *coverage* and *correctness*. By using accordance instead of either correctness or coverage, we avoid that the learning algorithm optimizes only correctness by leaving all difficult cases unclassified or optimizes coverage only by assigning a – possibly wrong – class label to each case.

3.2. Failure quality concepts

The failure quality concepts are complementary to the success quality concepts. When a classifier cannot classify all cases and/or misclassifies a certain amount of cases, these failure concepts reveal different causes for the deteriorated performance. There are three concepts that explain why cases remain unclassified: *omittance*, *interference* and *restrictedness*.

Omittance is defined as:

> The extent to which the cases remain unclassified because of missing or invalid input attribute values.

The reason for having *omittance* is not directly related to how the classifier works. *Omittance* reflects the fact that in medicine information is not always available; for example, a test might fail or a certain test may not be performed because it was deemed to be unethical or economically unfeasible to perform.

Interference is defined as:

> The extent to which a classifier cannot assign a class label to a case because there seems to be supportive evidence in the input attributes for more than one class.

For our NNs this occurs when more than one output node has an output value that exceeds the node's threshold (in our case 0.5 for each node). When such a situation occurs, the network is apparently not able to perform a proper mapping between the input values and the output classes. This may occur because the specific input pattern has characteristics of two or more classes or because the network is not able to partition properly the input space. The latter can be a result of either an insufficient number of hidden units or of insufficient training.

Interference is also a useful concept for analysing a classifier with multiple (linear) discriminants. Here, voting schemes are often used to combine the output of the various discriminants. Interference occurs when more than one class gets a relatively large number of votes.

Restrictedness is defined as:

> The extent to which a classifier would leave a case unclassified because it does not seem to resemble any of the descriptions of the available classes.

Restrictedness is just the opposite of *interference*. Here the input pattern of a case is so different from the input patterns of the cases in the learning set that the NN does not contain the mapping for that input pattern to the correct class. Here again, restrictedness is a concept rather specific for NN classifiers.

Also the *correctness* concept has a pair of accompanying concepts that describe the nature of the errors made by the classifier, i.e. *bias* and *dispersion*.

Bias is defined as

> The extent to which a classifier will have a preference to misclassify cases in one or a few categories.

Our definition of *bias* should not be confused with the bias concept in other domains, where it is used to indicate the systematic difference between (the expected value of) an observed value and the target value. The only relation of our concept with the others is the focus on the systematic aspect.

Dispersion is defined as

> the extent to which a classifier evenly distributes the misclassified cases over the categories.

Correctness can be related to the *accuracy* concept as used in e.g. biochemistry. Both express the tendency to provide the correct result. The *bias* and *dispersion* concept are both related to the *precision* concept [31]. *Bias* and *dispersion* as well as *precision* express the variability in the results. Whereas *precision* takes into account all observations, *bias* and *dispersion* are only describing how the misclassified cases are distributed in the contingency table; the table in which co-occurrences of class labels in the database and assigned class labels are tabulated (see Eq. (8)).

Bias in the classification results may be corrected by changing the thresholds that are used to interpret the output values of the nodes in the output layer. There is, however, no guarantee that a higher overall correctness will also be obtained.

3.3. Class conditional quality concepts

So far we have introduced quality concepts that describe the general properties of a classifier. In certain applications and situations, overlooking a certain class

may be costly. In others, one wants perhaps to reduce the misclassifications for one class as it involves a large number of cases. It may also be possible that a user wants to be sure that the assignment of a certain class label has a high likelihood of being correct as the economical and/or ethical costs of further investigations for those cases may be high. To assess the applicability of a classifier for one's own situation, the global quality concepts are insufficient.

For that reason we have also defined the *class conditional* variants of those concepts. The idea is that one does not express the extent to which that quality aspect is present for the whole set of cases but only for those cases that belong to a certain class.

For example, the *class conditional coverage* is defined as

> the extent to which cases from a certain class are assigned a class label by the classifier.

Such class conditional quality concepts are determined by only considering one column or one row in the contingency table.

The idea of class conditional quality concepts is often used in the evaluation of medical classification algorithms. For two-class classifiers *class conditional correctness* is often expressed as the *sensitivity* and *specificity*, [15]. However, these concepts do not take any unclassified cases into account.

Apart from quality concepts, which are conditional on the *class label in the database*, the *correctness*, *bias* and *dispersion* can also be defined conditional on the *class label assigned by the classifier*. The latter type of *class conditional correctness* is also known as the *predictive value of the outcome of a classifier*.

4. Quality metrics

In this section, we will define a set of metrics for measuring the extent to which a classifier has certain properties. *All proposed metrics are based on the information available in a contingency table.* We define a contingency table as follows:

$$M = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,c} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,c} \\ \cdots & \cdots & \cdots & \cdots \\ m_{c,1} & m_{c,2} & \cdots & m_{c,c} \\ m_{c+1,1} & m_{c+1,2} & \cdots & m_{c+1,c} \\ m_{c+2,1} & m_{c+2,2} & \cdots & m_{c+2,c} \\ m_{c+3,1} & m_{c+3,2} & \cdots & m_{c+3,c} \end{bmatrix} \tag{8}$$

The last three rows reflect the nonclassified cases, due to *omittance*, *interference* and *restrictedness*, respectively. The elements $m_{i,i}$, $i \leq c$, represent the correctly classified cases for class i.

An example contingency table is given in Table 1. Here the rows $c + 1$, $c + 2$ and $c + 3$ are merged into one row, representing the unclassified cases.

Table 1
An example contingency table for a classifier. The numbers appearing in the shaded cells represent the correctly classified cases

		Database			
Classifier		Class 1	Class 2	Class 3	Marginal
	Class 1	23	3	2	28
	Class 2	8	28	1	37
	Class 3	0	0	26	26
	Non. class.	2	2	4	8
	Marginal	33	33	33	99

The total number of cases that receive a class label, is defined by

$$s = \sum_{i=1}^{c} \sum_{j=1}^{c} m_{i,j} \tag{9}$$

Let the total number of unclassified cases be determined from

$$u = \sum_{i=c+1}^{c+3} \sum_{j=1}^{c} m_{i,j} \tag{10}$$

The coverage of a classifier is easily determined from

$$\Omega = \frac{s}{s+u} \tag{11}$$

The coverage measure Ω follows a binomial distribution because passing $s+u$ cases through a network may be seen as a sequence of $s+u$ trials with two outcomes possible for each case (it either is classified or remains unclassified), and the processing of each case is independent of the processing of other cases.
Omittance, interference and *restrictedness* are expressed as fractions of the unclassified cases:

$$\begin{aligned} \textit{omittance } \iota &= \frac{\sum_{j=1}^{c} m_{c+1,j}}{u} \\ \textit{interference } \phi &= \frac{\sum_{j=1}^{c} m_{c+2,j}}{u} \\ \textit{restrictedness } \psi &= \frac{\sum_{j=1}^{c} m_{c+3,j}}{u} \end{aligned} \tag{12}$$

The class conditional metrics for *coverage, omittance, interference* and *restrictedness* for the various classes in the database are defined in a similar way, by leaving out the summations over j in (9), (10) and (12).

For *correctness* several quality metrics are possible. A simple one is the fraction of correctly classified cases defined as

$$\rho = \frac{\sum_{k=1}^{c} m_{k,k}}{s} \tag{13}$$

Note that this fraction is taken relative to the number of cases that are assigned a class label by the classifier.

Also class conditional *correctness* metrics can be defined. Note that now we can condition the *correctness* on the *true class label* (equivalent to the *sensitivity* and *specificity* concepts) or the *assigned class label* (equivalent to the *predictive value* of a classification). These quality metrics will be denoted as ρ_i^D and ρ_i^C, respectively. [1] Let's define

$$s_i^D = \sum_{k=1}^{c} m_{k,i} \tag{14}$$

and

$$s_i^C = \sum_{k=1}^{c} m_{i,k} \tag{15}$$

Now the class conditional *correctness* metrics are defined as

$$\rho_i^D = \frac{m_{i,i}}{s_i^D} \tag{16}$$

and

$$\rho_i^C = \frac{m_{i,i}}{s_i^C} \tag{17}$$

All metrics defined so far estimate the probability of a binomial distribution. Hence, for any metric – say x – one can compute the standard error of the estimate $\hat{x}$:

$$\sigma_x = \sqrt{\frac{\hat{x}(1-\hat{x})}{n}} \tag{18}$$

with n being the number in the denominator of the formula for the metric. When both $\hat{x}n > 5$ and $(1-\hat{x})n > 5$ the confidence interval for x is given by

$$\hat{x} + \frac{\Phi_{\alpha/2}^2(0.5-\hat{x})}{n} - \Phi_{\alpha/2} \times \sigma_x \leq x \leq \hat{x} + \frac{\Phi_{\alpha/2}^2(0.5-\hat{x})}{n} + \Phi_{\alpha/2} \times \sigma_x \tag{19}$$

with $\Phi_{\alpha/2}$ denoting the value cutting off the area $\alpha/2$ in the upper tail of the

[1] Quality metrics and supporting variables, which are conditional on the true class label, are given the superscript D; those conditional on the assigned class label the superscript C.

standard normal distribution. When both $\hat{x}n > 50$ and $(1-\hat{x})n > 50$ the confidence interval for x can be approximated by

$$\hat{x} - \Phi_{\alpha/2} \times \sigma_x \leq x \leq \hat{x} + \Phi_{\alpha/2} \times \sigma_x \tag{20}$$

Another metric that is often used to quantify the degree of agreement between two observers, but which can also be used to characterize the *correctness* of a classifier, is the kappa (κ) metric [7]. The κ metric determines the degree of agreement exceeding the agreement by chance alone. This metric is defined as

$$\kappa = \frac{\rho - e}{1 - e} \tag{21}$$

with

$$e = \sum_{k=1}^{c} \frac{s_k^D \times s_k^C}{s^2} \tag{22}$$

It has been shown that the interpretation of the κ values is not unproblematic [11,13,42], specifically when the different classes have largely different a priori probabilities. In general, one can say that a κ-value above 0.75 indicates excellent agreement [11]. The standard error for κ is defined in [13,14].

Contrary to the widespread use of the κ-metric is the very sparse use of the class conditional kappas [24,28]. Let's define

$$e_i^D = \frac{s_i^D}{s} \tag{23}$$

and

$$e_i^C = \frac{s_i^C}{s} \tag{24}$$

The term e_i^D denotes the marginal probability that a case belongs to class i. It is used as the expected number of correctly classified cases in row i. The κ-metric. conditional on the assigned class label is given by

$$\kappa_i^C = \frac{\rho_i^C - e_i^D}{1 - e_i^D} \tag{25}$$

A similar formula can be given for the κ-metric conditional on the true class label. The standard error for the conditional κ-metric has been derived as [28]

$$\sigma_{\kappa_i^C} = \sqrt{\frac{\rho_i^C(1-\rho_i^C)}{s(1-e_i^D)^2}} \tag{26}$$

Defining metrics for the *bias* and *dispersion* quality concepts is more difficult. The overall *dispersion* concept can be interpreted as the degree to which misclassifications of true class i into class j are compensated by misclassifications of the true class j into class i. This compensation has to take into account the a priori

class probabilities. When ten cases of class A are classified as B and only one case of class B is classified as class A, then there seems not to be a compensation. However, when there are ten times as many cases in class A as compared to the number of cases in class B, then one can say that the misclassifications compensate each other completely.

To derive a metric for the overall *bias / dispersion* we need to normalize the contingency table such that all columns have the same number of cases. The number of cases belonging to class i is given by

$$R_i = \sum_{k=1}^{c+3} m_{k,i} \tag{27}$$

Define R' as the number of cases in the rarest class:

$$R' = \min\{R_i\}_1^c \tag{28}$$

The elements of the normalized contingency table M' are given by

$$m'_{i,j} = \frac{m_{i,j} \times R'}{R_i} \tag{29}$$

The overall *dispersion* of a classifier is measured as the degree of *symmetry* of the normalized contingency table M'. In [12] it was shown that

$$X = \sum_{j=1}^{c-1} \sum_{i=j+1}^{c} \frac{(m'_{j,i} - m'_{i,j})^2}{m'_{j,i} + m'_{i,j}} \tag{30}$$

follows a χ^2 distribution, provided that the denominator terms in (30) are large enough (at least 1 [33]). The overall *dispersion* is now defined as the likelihood that $m'_{i,j}$ and $m'_{j,i}$, $\forall j \neq i$, are equal:

$$\pi = \chi^2(X, df) \tag{31}$$

with $df = c(c-1)/2$ being the degrees of freedom.
The *bias* metric is defined as

$$\theta = 1 - \pi \tag{32}$$

The *bias* and *dispersion* metrics can also be defined conditional on the class labels. These conditional metrics describe different properties than the overall *bias* and *dispersion*. We assume for a dispersed classifier that if the number of cases in the classes is large enough the distribution or misclassified cases will follow the reduced marginal distribution of cases. If that is not so, there is some systematic tendency in the way the classifier misclassifies cases; the classifier is biased.
The reduced marginal distribution for class i is based on all cases in the database, *excluding the cases belonging to class i* (see Fig. 3). For class i, the reduced marginal probability follows from

$$P^D_{i,j} = \frac{R_j}{\sum_{k=1}^{c} R_k - R_i} \quad \forall j \neq i \tag{33}$$

	Class 1	Class 2	Class 3	Marginal
Class 1				
Class 2				
Class 3				
Unclass.				
Marginal	A	B	C	T
Red. M.		B/(T-A)	C/(T-A)	

Misclassified cases, class 1

Reduced marginal distribution, class 1

Fig. 3. An illustration of how the reduced marginal distribution of a class relates to the marginal distribution in a contingency table.

The number of misclassified cases among those given class label j is given by

$$n_j^C = \sum_{\substack{k=1 \\ k \neq j}}^{c} m_{j,k} \tag{34}$$

The expected number of misclassified cases with a true class label i follows from

$$\tilde{m}_{j,i} = n_j^C \times P_{i,j}^D \quad \forall i \neq j \tag{35}$$

Now we can compute

$$X_j^C = \sum_{\substack{i=1 \\ i \neq j}}^{c} \frac{\left(m_{j,i} - \tilde{m}_{j,i}\right)^2}{\tilde{m}_{j,i}} \tag{36}$$

X_j^C is χ^2-distributed and the class conditional *dispersion* can be defined as

$$\pi_j^C = \chi^2\left(X_j^C, df\right) \tag{37}$$

with $df = c - 2$. Accordingly, the class conditional *bias* follows from

$$\theta_j^C = 1 - \pi_j^C \tag{38}$$

Furthermore, the sign of

$$m_{j,i} - \tilde{m}_{j,i} \tag{39}$$

gives an indication whether a classifier is biased towards or away from class i when a class label j is erroneously assigned.

5. Experimental results

5.1. Clinical domain

We used a database of biochemical data on patients, who had been tested for thyroid functional disorders, as a basis for a set of experiments. The data were collected between 1981 and 1983 in the department of Clinical Chemistry, Copenhagen University Hospital at Hvidovre, Denmark. In total, the learning database consisted of 20 Hypothyroids (Myxoedema), 50 Hyperthyroids (Thyrotoxic) and 132 Euthyroids (normals) and controls.
In all our experiments we used the following attributes on which the NNs should diagnose the patients:

- the concentration of the triiodothyronine hormone (T3);
- the concentration of the thyroxine hormone (T4);
- the concentration of the thyroxine binding globulin (TBG) and
- the ability of serum to uptake radioactive T3.

In six cases (2 Hyperthyroids, 4 Euthyroids) measurements for TBG and/or T3-uptake were missing, resulting in an overall omittance ι of 3%.

We also had a test set of 174 cases available. In this set ι was 17%. Only 144 cases (12 Myxoedemas, 81 Euthyroids, 51 Thyrotoxics) had all four measurements available. We considered this set sufficiently large to assess the performance of the various networks we had generated from the learning set.

5.2. Factors of variation

As stated before, there are a number of design choices that influence the resulting NN. We have varied the following parameters independently from each other.

- The percentage of cases from the learning set used for training. We used 25, 35 and 50% of the 196 complete cases of the learning set. We grouped our experiments according to these fractions into three experiment series.
- We used for each fraction of cases from the learning set two different levels of *model complexity*, defined as the number of parameters (weights and biases) over the number of learning cases: 0.4 and 0.6 respectively.
- For each topology (defined by the model complexity), we determined three different sets of initial weights, each weight randomly generated in the range of $[-0.5, 0.5]$.
- For each fraction of cases, combined with a certain model complexity and set of initial weights, several subsets of cases were randomly chosen from the learning database. We did not pose any restriction on the class composition of the selected cases.

We considered a network properly trained on a subset of cases from the learning set, when it can classify correctly 96% of the training instances used (*accordance* $\geq$ 96%). Furthermore, we allowed the algorithm to cycle at most 2600

times through the set of selected learning cases. In case the network was unable to classify 96% of the training cases correctly after 2600 cycles, the network was discarded. In each experiment series, we trained 300 networks with an *accordance* $\geq 96\%$, resulting in a total of 1200 trained and converged networks.

These 1200 networks were all tested with the 144 cases of the independent test set. The obtained quality measures were used to test various hypotheses regarding the influence of the design criteria on the performance of the networks. The main conclusions are that

- networks do vary considerably regarding performance and therefore, model selection is *necessary*, and
- the composition of the set of training cases is the only factor from those varied in our experiments that influences performance systematically.

It was also shown that in the selection process, the user has to make a tradeoff between the various aspects of quality. Optimizing one of the quality measures will result in a less optional result for the others.

A few salient results warrant further discussion. First, we analyzed the reason cases could not be classified by an NN. It turned out that in nearly all generated networks the number of cases that could not be classified because of interference (more than one output node had an activation larger than the threshold) exceeded the number of cases due to restrictedness (no output node had an activation larger than the threshold).

Fig. 4 shows the distribution of the number of networks in which a certain number of cases could not be classified due to interference and due to restrictedness. The graph shows the results for the 300 networks that were trained with 50% (98) of the cases in the learning set. In the test set hardly any case could not be classified because it didn't fit the knowledge learned. So the set of training cases seems in general to reflect very well the distribution of the attribute values of the cases in real practice.

The bias/dispersion metrics were used to analyze the extent to which systematic misclassifications were made. It turned out, that both the Hypothyroid and the Hyperthyroid class had only misclassified Euthyroid cases. In almost all networks these two classes are significantly biased towards the Euthyroid class. This was to be expected from the nature of the domain. The three classes (Hypothyroid, Euthyroid and Hyperthyroid) form a kind of continuum, ranging from a decreased functioning to an increased functioning of the thyroid gland. Since the values of the attributes are more or less proportional to the level of functioning of the thyroid gland, misclassifications among the Hypo- and Hyperthyroid classes are very unlikely.

Already in 1977 Devijver showed that within a noisy domain the user must make a tradeoff between correctness and coverage [10]. More recently, a study showed that classifiers derived with various methods from the same database had different properties [35], which justifies the making of a tradeoff between the various quality aspects. For an NN classifier it can be expected that its coverage can be altered by changing the rule that is used to interpret the values of its output nodes. We expect that an increase in a classifier's coverage will lead to a drop in correctness.

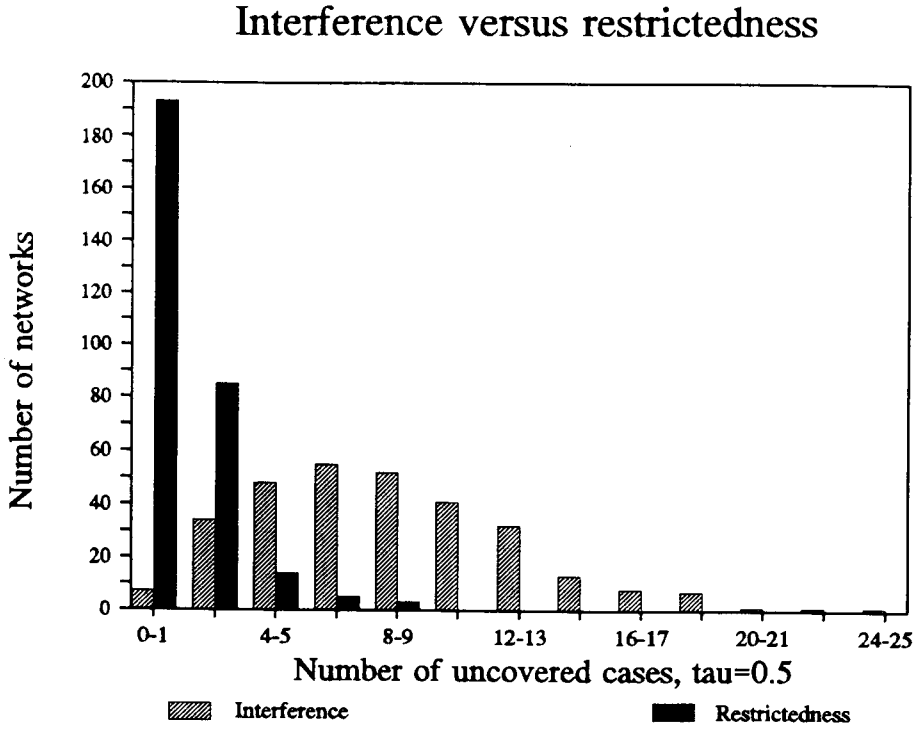

Fig. 4. The distribution of the number of networks that had a certain number of cases not classified because of interference and because of restrictedness. Results are shown for the 300 networks generated from random samples of 50% of the set of training cases. The threshold for the interpretation of the activation of the nodes in the output layer was set to 0.5 for each of the nodes ($\tau_i = 0.5$).

We investigated whether such a tradeoff also existed among a series of networks: do networks with a high coverage have a lower correctness and visa versa. It turned out that this is not generally true (see Fig. 5).

Networks that have a coverage in a certain range can have a large variation in their correctness measures. There is, however, a certain boundary beyond which no network exists. This observation makes it necessary that the users somehow express their preferences with respect to various quality concepts to allow for a multicriteria decision analysis on the quality measures of the generated networks. The users can make statements in terms of the amount of reduction in, for example, correctness they are willing to accept for a certain increase in another quality measure.

That this tradeoff not only can be made for correctness and coverage is shown in Fig. 6. In this scatterplot of correctness versus bias, three different preference profiles are shown. Preference 1 indicates that the user is willing to have a lower correctness, providing that the misclassifications are dispersed (the misclassifications of class A as class B are cancelled by approximately the same amount of

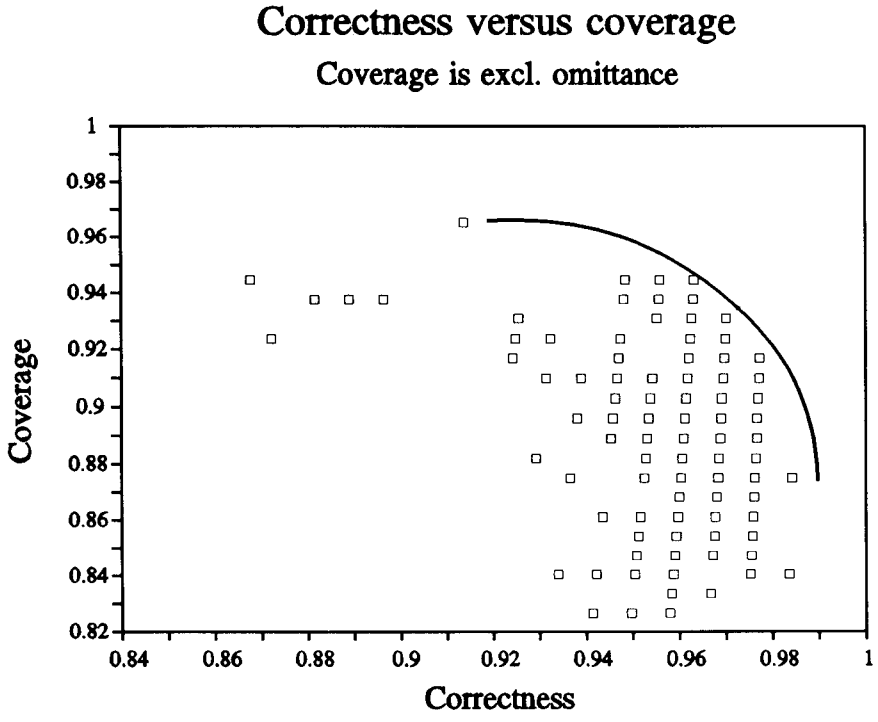

Fig. 5. A scatter plot of correctness versus coverage for a series of networks. There seems to be a boundary on the 'optimal' performance of a network, allowing for a tradeoff between correctness and coverage. Data are shown for the networks trained with 25% of the cases of the learning set. The curve indicates the limit which cannot be exceeded.

cases of class B that are classified as class A). Preference 2 indicates a user who just wants to have the best correctness. He does not mind whether the classifier is biased or dispersed. Preference 3 indicates the profile of a user who prefers a biased classifier and is willing to have a less correct classifier, provided it is more biased, but only to a certain extent.

Clearly the selection of an optimal NN classifier is not a trivial task. In fact, one needs a full multicriteria decision making support for finding the network(s) that best meets user needs. This is particularly true when the users want to express their quality preferences on the basis of the conditional quality metrics. For example, are they willing to have a lower correctness for class A when the coverage for class B increases, and to what extent? A thorough discussion of this issue is out of scope of this paper. It suffices here to indicate that our experimentations with a technique called 'Pareto race' [26] showed promising results. However, more experimentation and comparison with other methods like fuzzy sets [43] is needed, before a generally valid statement about the utility of this approach can be made.

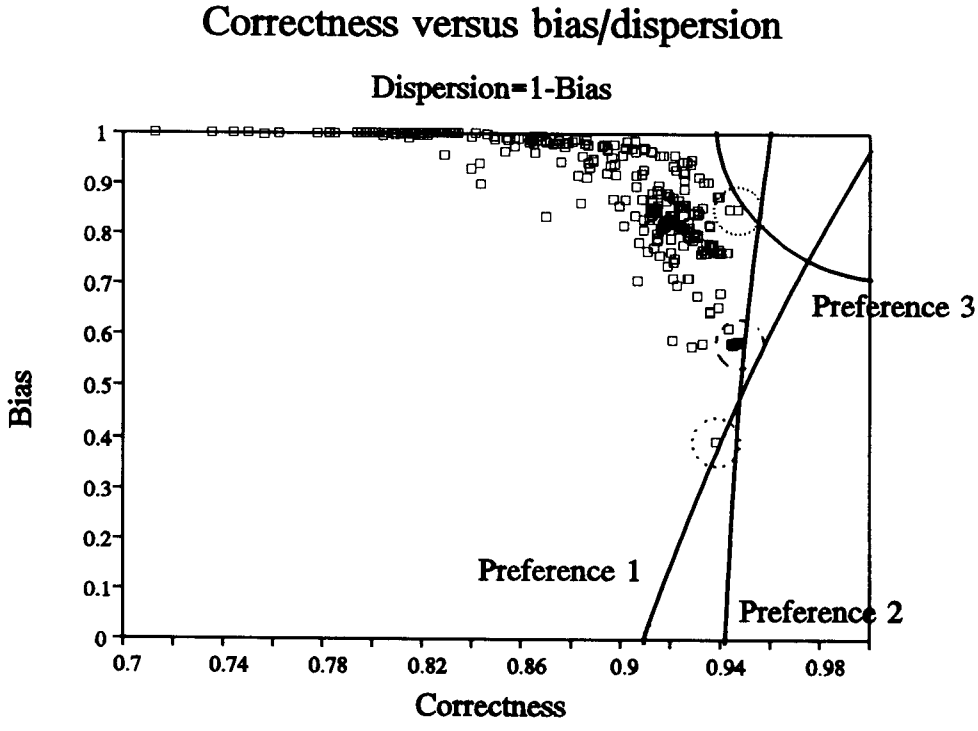

Fig. 6. A scatter plot of correctness versus bias, together with 3 different preference profiles. This figure shows the networks in which only T3 and T4 were utilized. Most of the networks were biased. The three curves illustrate how different users' indifference curves might look like.

6. Conclusions

In this paper we argued for the need of several quality measures to describe the properties of an NN classifier. We introduced a set of quality concepts that is part of a larger framework. We provided a set of metrics covering all quality concepts presented. It was shown that these quality concepts and the corresponding metrics allow users to define criteria that can be used to select a (small) subset of networks that fulfil their needs. The quality concepts presented here are the more basic ones. Among others, one needs to have insight into e.g. the robustness, the credibility, time dependency and ultimately the transferability of a (series of) networks before one will embark on using an NN routinely for a particular task.

Although we focused mainly on the application of the quality concepts and metrics for NNs, the concept of testing and quantifying various quality aspects holds for other types of classifiers as well.

Acknowledgements

The research reported in this paper has been performed in the framework of the KAVAS - Knowledge Acquisition, Visualization and Assessment Study - (A1021) and KAVAS-2 (A2019) projects. These projects have partially been funded by the Commission of the European Communities under the Exploratory phase of AIM and the current AIM Telematics in Health Care programme and by the Academy of the Technical Sciences, Denmark (EF-348). M. Egmont-Petersen performed the research when he was at CRI A/S, Birkerød and DASY, Copenhagen Business School, Copenhagen, Denmark as a PhD student.

References

[1] E.B. Andersen, N.E. Jensen and N. Kougsgaard, *Theoretical Statistics for Economists* (Academic Press, Copenhagen, 2nd ed. in Danish, 1984).

[2] E. Baum and D. Haussler, What size net gives a valid generalization?, *Neural Computat.* 1 (1989) 151–160.

[3] J. Brender, Information systems validation, I: A method for validation of functional aspects, Master thesis, No. 89-1-22, Institute of Computer Science, University of Copenhagen, Denmark, 1989.

[4] J. Brender, P. McNair, H. Raun, J. Nolan and S. Vingtoft, Meta-knowledge as a means for quality management in knowledge-based systems, in: R. O'Moore, S. Bengtsson, J.R. Bryant and J.S. Bryden eds., *Proc. Medical Informatics Europe '90, Lecture Notes in Medical Informatics 40* (Springer Verlag, Berlin, 1990) 360–367.

[5] J. Brender, P. McNair, H. Raun, J. Nolan and S. Vingtoft, Meta-knowledge Concepts, Deliverable 24, Technical report META-1.1 of the KAVAS (A1021) AIM project, 2nd ed., 1990.

[6] J. Brender, J.L. Talmon and P. McNair, Framework for validation of semantic aspects of knowledge, in preparation.

[7] J.A. Cohen, A coefficient of agreement for nominal scales, *Educational and Psychological Measurement* 20 (1960) 37–46.

[8] J. Cunningham and S. Haykin, Neural network detection of small moving radar targets in an ocean environment, in: S.Y. Kung, F. Fallside, J.A. Sorenson and C.A. Kaufmann, eds., *Proc. 1992 IEEE Workshop on Neural Networks for Signal Processing* (IEEE, Piscataway, NJ, 1992) 306–315.

[9] C.-H. de Verdier, T. Aronsson and A. Nyberg, eds., Quality control in clinical chemistry – efforts to find an efficient strategy *Scand. J. Clin. Lab. Invest.* 44, suppl 172 (1984) 1–241.

[10] P.A. Devijver, Reconnaissance des formes par la méthode des plus proches voisins, report R346, Phillips Research Laboratories, Brussels, 1977.

[11] D. Donker, Interobserver variation in the assessment of fetal heart rate recordings, Doctoral dissertation, VU University Press, Amsterdam, 1991.

[12] B.S. Everitt, *Analysis of Contingency Tables* (Chapman&Hall, London, 1977).

[13] J.L. Fleiss, *Statistical Methods for Rates and Proportions* (John Wiley&Sons, New York, 2nd ed. 1981).

[14] J.L. Fleiss, L. Cohen and B.S. Everitt, Large sample standard errors of kappa and weighted kappa, *Psychological Bull.* 72 (1969) 323–327.

[15] W. Gerhardt and H. Keller, Evaluation of test data from clinical studies, I: Terminology, graphic interpretation, diagnostic strategies and selection of sample groups, II: Critical review of the concepts efficiency, receiver operated characteristics (ROC) and likelihood ratios, *Scand. J. Clin. Lab. Invest.* 46, suppl 181 (1986) 1–74.

[16] E.S. Gelsema, Pattern recognition and artificial intelligence in medical research and practice, *Methods of Informat. in Med.* 28 (1989) 63–65.
[17] L.K. Hansen and P. Salamon, Neural Network Ensembles, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 12 (1990) 993–1001.
[18] L.K. Hansen, C. Liisberg and P. Salamon, Ensemble methods for handwritten digit recognition, in: S.Y. Kung, F. Fallside, J.A. Sorenson and C.A. Kaufmann, eds., *Proc. 1992 IEEE Workshop on Neural Networks for Signal Processing* (IEEE, Piscataway, NJ, 1992) 333–342.
[19] S.J. Hanson and D.J. Burr, What connectionist models learn: Learning and representation in connectionist networks, *Behavioral and Brain Sci.* 13 (1990) 471–517.
[20] R.F. Harrison, S.J. Marshall and R.L. Kennedy, A connectionist aid to the early diagnosis of myocardial infarction, in: M. Stefanelli, A. Hasman, M. Fieschi and J. Talmon, eds., *Proc. Third Conf. on Artificial Intelligence in Medicine, Lecture Notes in Medical Informatics 44* (Springer Verlag, Berlin, 1991) 119–128.
[21] A. Hart and J. Wyatt, Connectionist models in medicine: an investigation of their potential, in: J Hunter, J. Cookson and J. Wyatt, eds., *Proc. AIME-89 Conf. Lecture Notes in Medical Informatics 38* (Springer Verlag, Berlin, 1989) 115–124.
[22] G.E. Hinton, Connectionist learning procedures, *Artificial Intelligence* 40 (1989) 185–234.
[23] G. Hripcsak, Using connectionistic modules for decision support, *Methods of Information in Med.* 29 (1990) 167–181.
[24] G.F. Jensen, P. McNair, J. Boesen and V. Hegedüs, Validity in diagnosing Osteoporosis, *Europ. J. Radiol.* 4 (1984) 1–3.
[25] A.N. Komolgorov, On the representation of continuous functions of several variables by superposition of continuous functions of a smaller number of variables, *Dokl. Akad.* 108 (1956) 179–182.
[26] P. Korhonen and J. Wallenius, A Pareto race, *Naval Research Logistics* 35 (1988) 615–623.
[27] S.Y. Kung, F. Fallside, J.A. Sorenson and C.A. Kaufmann, eds., *Neural networks for Signal Processing II, Proc. 1992 IEEE Workshop on Neural Networks for Signal Processing* (IEEE, Piscataway, NJ, 1992).
[28] R.J. Light, Measures of response agreement for qualitative data: Some generalizations and alternatives, *Psychol. Bull.* 76 (1971) 365–377.
[29] F.A. Lodewyk and E. Barnard, Avoiding false local minima by proper initialization of connections, *IEEE Trans. Neural Networks* 3 (1992) 899–905.
[30] M. Minsky and S. Papert, *Perceptrons* (MIT Press, Cambridge, MA, 1969).
[31] R.A. Nadkarni, The quest for quality in the laboratory, *Analytical Chemistry* 63 (1992) 675–682.
[32] P. Nykänen, ed., Issues in evaluation of computer-based support to clinical decision making, Research Report 127, Institute of Informatics, Oslo University, 1989.
[33] T. Read, R.C. Noal and A.C. Cressie, *Goodness-of-fit Statistics for Discrete Multivariate Data* (Springer Verlag, New York, 1988).
[34] D.E. Rumelhart, J.L. McClelland and the PDP Research Group, *Parallel Distributed Processing, Explorations in the Microstructure of Cognition, Vol 1: Foundations* (MIT Press, Cambridge, MA, 1986).
[35] T. Schiøler, W. Grimson, P. Sharpe, M. Egmont-Petersen, G. Momsen, R. O'Moore and P. McNair, Automatic decision support based on voting by independent decision support systems, *Proc. Computing in Clinical Laboratories '92* (1992) 58.
[36] C.N. Schizas, C.S. Pattchis, T.S. Schofield, and P.R. Fawcett, Artificial Neural Nets in computer-aided macro motor unit potential classification, *Trans. IEEE Eng. in Med. and Biol.* (1990) 31–38.
[37] J.W. Shavlik and G.G. Towell, An approach to combining explanation-based and neural learning algorithms, *Connection Sci.* 1 (1989) 231–252.
[38] S. Siegel, *Nonparametric Statistics for the Behavioral Sciences* (McGraw-Hill, Kogakusha, Tokyo, 1956).
[39] P.K. Simpson, *Artificial Neural Systems, Foundations, Paradigms, Applications and Implementations* (Pergamon Press, New York, 1990).
[40] G.G. Towell and J. Shavlik, Extracting refined rules from knowledge-based neural networks, *Machine Learning* 13 (1993) 71–101.
[41] F. Vogelsang, Segmentierung radiologisch dokumentierter fokaler Knochenläsionen auf Basis

kontextbezogener Vektoren mit neuronalen Netzwerken, Diplomarbeit (Master Thesis), Fakultät für Informatik, Medizinische Fakultät, RWTH Aachen, 1993.

[42] J.L. Willems, C. Abreu-Lima, P. Arnaud, C.R. Brohet, B. Denis, J. Gehring, I. Graham, G. van Herpen, H. Machado, J. Michaelis and S.D. Moulopoulos, Evaluation of ECG interpretation results obtained by computer and cardiologists, *Methods of Informat. in Med.* 29 (1990) 308–316.

[43] L.A. Zadeh, Making computers think like people, *IEEE Spectrum* (Aug. 1984) 26–32.