On the misuses of artificial neural networks for prognostic and diagnostic classification in oncology

Guido Schwarzer\textsuperscript{1,*\textsuperscript{,†}, Werner Vach\textsuperscript{2} and Martin Schumacher\textsuperscript{1}}

\textsuperscript{1}University of Freiburg, Institute of Medical Biometry and Medical Informatics, Stefan-Meier-Straße 26, D-79104 Freiburg, Germany
\textsuperscript{2}University of Southern Denmark/Odense University, Department of Statistics and Demography, Campusvej 55, DK-5230 Odense M, Denmark

SUMMARY

The application of artificial neural networks (ANNs) for prognostic and diagnostic classification in clinical medicine has become very popular. In particular, feed-forward neural networks have been used extensively, often accompanied by exaggerated statements of their potential. In this paper, the essentials of feed-forward neural networks and their statistical counterparts (that is, logistic regression models) are reviewed. We point out that the uncritical use of ANNs may lead to serious problems, such as the fitting of implausible functions to describe the probability of class membership and the underestimation of misclassification probabilities. In applications of ANNs to survival data, further difficulties arise. Finally, the results of a search in the medical literature from 1991 to 1995 on applications of ANNs in oncology and some important common mistakes are reported. It is concluded that there is no evidence so far that application of ANNs represents real progress in the field of diagnosis and prognosis in oncology. Copyright © 2000 John Wiley & Sons, Ltd.

1. INTRODUCTION

During the last years, the application of artificial neural networks (ANNs) for prognostic and diagnostic classification in clinical medicine has attracted growing interest in the medical literature. For example, a ‘mini-series’ on neural networks that appeared in the \textit{Lancet} contained three more or less enthusiastic review articles [1–3] and an additional commentary expressing some scepticism [4]. In this commentary as well as in other, more balanced reviews [5], the comparison with competing statistical methods is emphasized.

The relationship between ANNs and statistical methods, especially logistic regression models, has been described in several articles [6–10]. We start with a brief summary of feed-forward neural networks and logistic regression models and their extensions. We then illustrate by means...
of a simple example that uncritical use of ANNs can lead to functions describing the probability of class membership that are far from plausible. This is due to the flexibility of ANNs which is often cited as an advantage [2]; we argue that it must be seen as a major concern. Problems associated with the estimation of misclassification probabilities and the application of ANNs to survival data that often occur in oncology are outlined. In addition to these general methodological considerations we report the results of a literature search on the application of ANNs in oncology; the most frequently occurring mistakes are identified and commented on. Finally, an attempt is made to summarize our findings in the conclusions section.

2. ESSENTIALS OF LOGISTIC REGRESSION AND NEURAL NETWORKS

The common principle of diagnosis and prognosis in oncology or other areas is to decide to which class \( k, k \in \{0, \ldots, K\} \), an individual belongs by using information on a set of \( p \) covariate values \( x = (x_1, \ldots, x_p)' \). The simplest but most common case is to consider only two possible classes. Examples are the diagnosis of a tumour as ‘benign’ or ‘malignant’ or the prediction of the recurrence of a disease up to a specified point in time. We restrict our attention to the special case of two classes and call the class indicator \( > M_0, 1 \).

The aim is to construct a decision rule for individuals with known covariate values but unknown class level. Many approaches to constructing such a decision rule are based on estimating the conditional probability of observing an individual with class level 1 given the covariates \( x = (x_1, \ldots, x_p)' \). Such an estimation is often based on a parametric model

\[
P(Y = 1 \mid X = x) = f(x, \beta)
\]

for the conditional probabilities, where \( \beta \) is a vector of unknown parameters. The parameters are called ‘regression coefficients’ in statistics and ‘weights’ by the neural net community. Estimation of these parameters is done by using information from a sample of individuals with known class levels and known covariate values. Based on an estimate \( \hat{\beta} \) of the parameters, we can estimate for each future individual with covariate values \( x \) the conditional probability of belonging to class 1 by \( f(x, \hat{\beta}) \). The classification rule is given as: class 1, if \( f(x, \hat{\beta}) \) is greater than some cut-off level, for example, 0.5; class 0, otherwise.

Logistic regression and feed-forward neural networks differ in the functional form \( f(x, \beta) \) assumed for the conditional probability \( P(Y = 1 \mid X = x) \).

2.1. Logistic regression

The linear logistic regression model [11] assumes that

\[
f(x, \beta) = \frac{1}{1 + \exp(-\beta_0 - \sum_{i=1}^{p} \beta_i x_i)} = \Lambda \left( \beta_0 + \sum_{i=1}^{p} \beta_i x_i \right)
\]

with \( \Lambda () \) denoting the logistic function. The model can be re-expressed in terms of the odds of observing an individual with class level 1 given the covariates \( x \) in the following way:

\[
\text{Odds}(x) = \frac{P(Y = 1 \mid X = x)}{P(Y = 0 \mid X = x)} = \exp \left( \beta_0 + \sum_{i=1}^{p} \beta_i x_i \right).
\]
Thus the essential assumption of the linear logistic regression model is that the logarithm of the odds is linear in \( x \). The odds ratio of two individuals, showing a difference of \( \Delta \) for the \( i \)th covariate and no difference for the others, is equal to \( \exp(\beta_i \Delta) \). Hence, the regression coefficients \( \beta_i \) are directly interpretable as log-odds ratios or, in terms of \( \exp(\beta_i) \), as odds ratios. This property of the logistic regression model enables one to assess the effect of a single covariate and is one major reason for the widespread use of this method in the biomedical sciences.

A natural extension of the linear logistic regression model is to include quadratic terms and multiplicative interaction terms:

\[
f(x, \beta) = \Lambda \left( \beta_0 + \sum_{i=1}^{p} \beta_i x_i + \sum_{i=1}^{p} \gamma_i x_i^2 + \sum_{i<j} \delta_{ij} x_i x_j \right).
\] (3)

These more general logistic regression models are usually considered as the first step to overcome the stringent assumption of additive and purely linear effects of the covariates. Other approaches to model the conditional probability \( P(Y = 1 \mid X = x) \) in a more flexible manner include generalized additive models [12], fractional polynomials [13] and feed-forward neural networks.

2.2. Feed-forward neural networks

In articles published in biomedical journals, ANNs are typically introduced as a computer-based procedure functioning in a manner analogous to the human brain. The description is often accompanied by a graphical representation like that in Figure 1. This figure illustrates a feed-forward neural network with one hidden layer. The number of hidden layers may vary but is almost always chosen to be one. The network consists of \( p \) input units, \( r \) hidden units and one output unit. The flow of information is indicated by the arrows.

The units in the input layer \( x = (x_1, \ldots, x_p) \) correspond to the covariates in a logistic regression model. The hidden units \( h_1, \ldots, h_r \) are the result of applying a so-called ‘activation function’ to a weighted sum of the input units plus a constant which is called ‘bias’ in the neural
net framework. The value of a hidden unit \( h_j \) is given by

\[
h_j = \phi_j^{\eta} \left( w_{0j} + \sum_{i=1}^{p} w_{ij} x_i \right)
\]

with unknown parameters \( w_{0j}, \ldots, w_{pj} \). Although the activation functions \( \phi_j^{\eta}(\cdot) \) could be mutually different they are almost always chosen as the logistic function \( \Lambda(\cdot) \). Thus the value of a hidden unit appears to be

\[
h_j = \frac{1}{1 + \exp(-w_{0j} - \sum_{i=1}^{p} w_{ij} x_i)} = \Lambda \left( w_{0j} + \sum_{i=1}^{p} w_{ij} x_i \right).
\]

The value of the output unit \( y \) is calculated by applying another activation function \( \phi^{\eta}(\cdot) \) to the weighted sum of the values of the hidden units plus bias. The output activation function must not coincide with any \( \phi_j^{\eta}(\cdot) \) but in most applications it is also chosen as the logistic function. Hence

\[
y = f(x, w) = \Lambda \left( W_0 + \sum_{j=1}^{r} W_j h_j \right) = \Lambda \left( W_0 + \sum_{j=1}^{r} W_j \Lambda \left( w_{0j} + \sum_{i=1}^{p} w_{ij} x_i \right) \right)
\]

with unknown weights \( w = (W_0, \ldots, W_r, w_{01}, \ldots, w_{pr}) \), which have to be estimated in a ‘learning’ or ‘training’ process. This representation of a feed-forward neural network may not have the appeal of Figure 1, but is the basis for analytical comparisons of feed-forward neural nets and logistic regression models.

One drawback of feed-forward neural networks is that the weights \( w \) have no clear interpretation. Except for a neural net without a hidden layer and with a logistic activation function, the weights lack the interpretation as log-odds ratios. In this special case formula (4) reduces to formula (2) which is a linear logistic regression model. This specific network type is called ‘logistic perceptron’ in the neural network literature. An extension of model (4) for \( K > 2 \) unordered categories is a feed-forward neural network with \( K \) output units and a softmax activation function [14,15] in the output layer. Such a network without a hidden layer is equivalent to use of a polytomous logistic regression model [16,17] for the conditional probabilities \( P(Y = k | X = x) \), \( k = 1, \ldots, K \). A feed-forward neural network with \( p \) input units, \( r \) hidden units and \( K \) output units is denoted by \( p-r-K \) in the following.

### 2.3. Estimation

A sample of \( N \) individuals with known covariate values \( x^{(n)} = (x_1^{(n)}, \ldots, x_p^{(n)}) \) and known class level \( y^{(n)} \in \{0, 1\} \) is used to estimate regression coefficients \( \beta \) or weights \( w \), respectively. ‘Training a network’ is equivalent to estimation.

In neural net science, estimation is usually based on the minimization of a distance measure called ‘energy function’ or ‘learning function’. In most applications the distance measure chosen is the least squares criterion:

\[
D_{LS}(w) = \sum_{n=1}^{N} (y^{(n)} - f(x^{(n)}, w))^2.
\]
Especially for classification problems, the Kullback–Leibler distance [18] has been proposed in the neural net literature [19,20] as an alternative:

$$D_{KL}(w) = \sum_{n=1}^{N} \left[ y^{(n)} \log \frac{y^{(n)}}{f(x^{(n)}, \theta)} + (1 - y^{(n)}) \log \frac{1 - y^{(n)}}{1 - f(x^{(n)}, \theta)} \right],$$ (6)

In statistical science, estimation is usually based on the maximum likelihood principle [21]. It is well-known that fitting neural networks or logistic regression models by the maximum likelihood principle is equivalent to minimization of the Kullback–Leibler distance [8].

Standard numerical optimization algorithms, like quasi-Newton or Levenberg–Marquardt algorithms, can be used to minimize the chosen distance measure. However, for the training of neural networks other algorithms are preferred. In particular, the back-propagation algorithm or one of its variants is implemented in most commercial neural net programs and thus utilized by the (biomedical) users of this software. The algorithm is known to suffer from convergence problems and difficulties in choosing values for the tuning-parameters embedded in this algorithm (‘learning rate’, ‘momentum’ etc.) [22,23]. The literature on numerical optimization provides many simple alternatives like Newton–Raphson methods allowing maximization of (5) or (6) within a few steps and the automatic choice of the tuning parameters.

3. IMPLAUSIBLE FUNCTIONS FITTED BY NEURAL NETWORKS

Feed-forward neural networks with one hidden layer are universal approximators [24–26] and thus can approximate any function defined by the conditional probability (1) with arbitrary precision by increasing the number of hidden units. However, the question in practice is whether we can expect valid approximations to the true function \( f \) by solving (5) or (6) based on the limited data available in biomedical applications. To examine this question we consider a simple example that was first presented by Finney [27] and has extensively been used for illustrative purposes [28–30].

The data set consists of a class indicator \( Y \in \{0, 1\} \) and two covariates \( X_1 \) and \( X_2 \) measured on 39 individuals. A scatter plot of the data displayed in Figure 2 indicates that, when considering both covariates, the two classes are well separated, except for two individuals marked in Figure 2. However, for each single covariate the separation is not as obvious.

In the single plots of Figure 3 we included the pairs \((Y, X_2)\), and we observe that the ranges of values of \( X_2 \) for \( Y = 0 \) and \( Y = 1 \), respectively, clearly overlap. For small values of \( X_2 \) the probability of \( Y = 1 \) is small but it increases with increasing \( X_2 \). We first fit a linear logistic regression model to this data, using the maximum-likelihood principle as provided by PROC LOGISTIC in SAS 6.11 [31]. The resulting estimated function \( x \mapsto f(x, \hat{\beta}) \) is shown in the first plot; it is a smooth and hence plausible proposal for the true \( f \).

Next we fit feed-forward neural networks with logistic activation functions by minimizing the Kullback–Leibler distance (6), using the back-propagation algorithm provided by the public domain software package Stuttgart Neural Network Simulator (SNNS) [32], version 4.1, with a self-written extension for the Kullback–Leibler criterion. Using a simple logistic perceptron, we achieve of course the same function \( x \mapsto f(x, \hat{\theta}) \) as in the case of linear logistic regression. Using neural networks with a hidden layer, we find ourselves confronted with functions becoming more and more warped with increasing numbers of hidden units. Such functions are increasingly
implausible, because the true function $f$ should reflect some biological or medical relationship that can assumed to be smooth, not warped.

Mathematically, the extreme fluttering of the functions shown in Figure 3 is due to some very large weights. This undesirable feature is well recognized in the literature on neural nets, and there exists the suggestion of introducing some weight decay [7] to overcome this problem, that is, to add the sum of squared weights to the distance measure $D(w)$ as penalty term

$$D^*(w) = D(w) + \lambda \left( \sum_{j=1}^{r} W_j^2 + \sum_{j=1}^{r} \sum_{i=1}^{p} w_{ij}^2 \right)$$

with $D(w)$ according to (5) or (6). The smoothness of the resulting function is controlled by the decay parameter $\lambda$. The effect of using weight decay is illustrated in the lower part of Figure 3. A neural network with 15 hidden units and $\lambda = 0.005$ results in a fit comparable to that of to the logistic regression model. However, weight decay seems to have rarely been used in biomedical applications.

The tendency of neural nets to fit implausible functions can also be illustrated by considering both covariates. Fitting a linear logistic model results in a function (Figure 4) which is still smooth, but clearly indicates the difference between the lower left and upper right part of Figure 2. The two observations printed in bold font in Figure 2 are here regarded as indicators that the change from the upper right part to the lower left part is not a change from a region with definite membership of class 1 to a region with definite membership of class 0, but that there is an
Figure 3. Estimated functions $x \mapsto f(x, \hat{\beta})$ for the probability of belonging to class 1 from the linear logistic regression model and from feed-forward neural networks with different number $r$ of hidden units and logistic activation function for covariate $X_2$ of the Finney data represented in Figure 2.

Figure 4. Grey scale image plots of the estimated function $x \mapsto f(x, \hat{\beta})$ for the probability of belonging to class 1 of linear logistic regression and neural networks analyses for both covariates of the Finney data represented in Figure 2. Darker grey scale levels represent higher probabilities of membership in class 1; displayed grey scale levels range from 0 to 1.
intermediate region where both memberships are possible. Feed-forward neural networks with one to five hidden units were fitted to the data. A feed-forward neural network with three hidden units fits almost perfectly but fails to recognize the intermediate region. Instead, a strict frontier between class 1 and class 0 membership is postulated, but this frontier is not smooth and hence barely plausible.

4. SURVIVAL DATA AND NEURAL NETWORKS

The application of feed-forward neural nets to survival data has recently been discussed [33–39]. It is fairly straightforward [37] to consider a ‘grouped’ version of the Cox regression model [40] in the framework of this paper. Denoting by $T$ the survival time random variable and by $I_k$ the time interval $t_{k-1} \leq t < t_k$, $k = 1, \ldots, K$, where $0 = t_0 < t_1 < \cdots < t_K < \infty$, the model can be specified through the conditional probabilities

$$P(T \in I_k \mid T \geq t_{k-1}, x) = \Lambda \left( \beta_{0k} + \sum_{i=1}^{p} \beta_{ik} x_i \right)$$

for $k = 1, \ldots, K$. This is the original proposal by Cox [40]; other models for grouped survival data can be obtained using other link functions [41,42]. The neural network corresponding to (7) is an extension of model (4) without a hidden unit to $K$ output units $y_1, \ldots, y_K$. The output $y_k$ at the $k$th output unit corresponds to the conditional probability of dying in the $k$th time interval $I_k$, $k = 1, \ldots, K$.

Data for the $n$th individual consists of a vector $x^{(n)} = (x_1^{(n)}, \ldots, x_p^{(n)})'$ of regressor variables and a vector $y^{(n)} = (y_1^{(n)}, \ldots, y_K^{(n)})'$ where $y_k^{(n)}$ is an indicator for individual $n$ dying in the interval $I_k$ and $K_n \leq K$ is the number of intervals in which individual $n$ is observed. Thus $y_1^{(n)}, \ldots, y_{K_n-1}^{(n)}$ are all zero and $y_{K_n}^{(n)}$ is equal to 1 if this individual died in $I_{K_n}$ and equal to 0 if he/she was censored. This situation implies that the network has a randomly varying number of output nodes according to those time intervals where an individual is ‘at risk’. This is a standard problem of survival analysis [43] and can easily be accommodated by using a slight modification of distance measure (6).

In particular, setting

$$y_k = f(x, w) = \Lambda \left( \beta_{0k} + \sum_{i=1}^{p} \beta_{ik} x_i \right)$$

where $w$ summarizes $\beta_{01}, \ldots, \beta_{0K}, \beta_{11}, \ldots, \beta_{pK}$, we obtain

$$D^{KL}_k(w) = \sum_{n=1}^{N} \sum_{k=1}^{K_n} \left[ y_k^{(n)} \log \frac{y_k^{(n)}}{f(x^{(n)}, w)} + (1 - y_k^{(n)}) \log \frac{1 - y_k^{(n)}}{1 - f(x^{(n)}, w)} \right].$$

This could also be written as the sum over all $K$ output units by introducing an indicator for individual $n$ being at risk at the $k$th time interval $I_k$, $k = 1, \ldots, K$, as it is usually done in survival analysis [44]. The proportional hazards assumption can then be implemented in (7) through the constraints $\beta_{ik} = \beta_i$ for $i = 1, \ldots, p$ and $k = 1, \ldots, K$ whereas the parameters $\beta_{0k}$ are allowed to differ for $k = 1, \ldots, K$. 

There have been some other unsound proposals for analysing grouped survival data by means of a feed-forward neural net. In order to predict outcome (death or recurrence) of individual breast cancer patients, Ravdin and Clark [38] and Ravdin et al. [39] use a network with only one output unit but using the number \( k \) of the time interval as additional input. Moreover, they consider the unconditional survival probability of dying before \( t_k \) rather than the conditional one as output. Their underlying model then reads

\[
P(T < t_k | x) = \Lambda \left( \beta_0 + \sum_{i=1}^{p} \beta_i x_i + \beta_{p+1} k \right)
\]

for \( k = 1, \ldots, K \). This parameterization ensures monotonicity of the survival probabilities but also implies a rather stringent and unusual shape of the survival distribution, since in the case that no covariates are considered (8) reduces to

\[
P(T < t_k) = \Lambda(\beta_0 + \beta_{p+1} k)
\]

for \( k = 1, \ldots, K \). Obviously, the survival probabilities do not depend on the length of time intervals \( I_k \), which is a rather strange and undesirable feature. Including a hidden layer in (8) is a straightforward extension retaining all the features summarized above. De Laurentiis and Ravdin [34] call such type of neural networks 'time-coded models'.

Another form of neural networks that has been applied to survival data are the so-called 'single time-point models' [34]. Since they are identical to a logistic perceptron or a feed-forward neural network with a hidden layer (4) they correspond to fitting of logistic regression models (2) or their generalizations (3) to survival data. In practice, a single time-point \( t \) is fixed and the network is trained to predict the \( t \)-year survival probabilities. The corresponding model is given by

\[
P(T < t | x) = \Lambda \left( \beta_0 + \sum_{i=1}^{p} \beta_i x_i \right)
\]

or its generalization when introducing a hidden layer. This approach is used by Burke [33] to predict 10-year survival of breast cancer patients based on various patient and tumour characteristics at time of primary diagnosis. McGuire et al. [45] utilized this approach to predict 5-year disease-free survival of patients with axillary node-negative breast cancer based on seven potentially prognostic variables. Kappen and Neijt [36] used it to predict 2-year survival of patients with advanced ovarian cancer [46] obtained from 17 pre-treatment characteristics. The neural network they actually used reduced to a logistic perceptron.

Of course, such a procedure can be repeatedly applied for the prediction of survival probabilities at fixed time-points \( t_1 < t_2 < \cdots < t_K \). For example, Kappen and Neijt [36] trained several \( (K = 6) \) neural networks to predict survival of patients with ovarian cancer after 1, 2, \( \ldots, 6 \) years.

The corresponding model reads

\[
P(T < t_k | x) = \Lambda \left( \beta_{0k} + \sum_{i=1}^{p} \beta_{ik} x_i \right)
\]

in the case that no hidden layer is introduced.

Note that without restriction on the parameters such an approach does not guarantee that the probabilities \( P(T < t_k | x) \) increase with \( k \), and hence may result in life-table estimators suggesting...
non-monotone survival functions. Closely related to such an approach are the so-called ‘multiple
time-point models’ [34] where one neural network with \( K \) output units with or without a hidden
layer is used.

The common drawback of these naive approaches is that they do not allow one to incorporate
censored observations in a straightforward manner, which is closely related to the fact that they
are based on unconditional survival probabilities instead of conditional survival probabilities as
the Cox model. Neither omission of the censored observations – as suggested by Burke [33] – nor
treating censored observations as uncensored are valid approaches, but a serious source of bias,
which is well-known in the statistical literature. De Laurentiis and Ravdin [34] propose to
impute estimated survival probabilities for the censored cases from a Cox regression model, that
is, they use a well established statistical procedure just to make an artificial neural network work.
Future developments should be based on the considerations given by Liestøl et al. [37] where the
standard requirements for the analysis of survival data are incorporated. Biganzoli et al. [47], for
example, use a similar approach. Faraggi and Simon [35] use the regression model arising from
a neural network with a hidden layer to define an extension of Cox’s proportional hazards
regression model that is applied to the well-known prostate cancer data [48].

5. ESTIMATION OF MISCLASSIFICATION PROBABILITIES

In constructing a new rule for prognosis or diagnosis, it is necessary not only to know the new
rule, but also to know its accuracy. A simple, meaningful and popular measure to describe the
accuracy of a rule is its misclassification probability, that is, the probability of classifying future
objects incorrectly. An obvious estimate of the misclassification probability is the apparent error
rate defined as the relative frequency of incorrect classifications if the new rule is applied in the
current sample, that is, in that sample we have used to construct the rule. However, this rate tends
seriously to underestimate the true misclassification probability, because the rule is made as
similar as possible to the current data, not to future data. This is especially true for classification
rules developed by neural networks where the tendency to overfitting constitutes a major
problem. To avoid this underestimation of the true misclassification probability, a widespread
 technique is to split the sample randomly into a learning set and a test set. The learning set is used
to construct the new rule, and the test set is used to estimate the misclassification probability by
applying the new rule and counting the incorrect classifications. This test set based error rate is
unbiased.

We illustrate the difference between the apparent error rate and the test set based error rate by
a small artificial example. We consider a constellation with five covariates, which are indepen-
dently and identically distributed as standard Gaussian. The probability of an object belonging to
one of the classes \( Y = 0 \) or \( Y = 1 \) depends only on the first two covariates, and is given by

\[
P(Y = 1 \mid X = x) = \begin{cases} 
0.85 & \text{if } x_1 > 0 \text{ and } x_2 > 0 \\
0.15 & \text{otherwise.}
\end{cases}
\]

This reflects a possible situation in medical applications: if two specific variables both exceed
a given threshold, we have a high probability that the object belongs to class \( Y = 1 \), otherwise this
probability is small. We draw a sample of size 400 according to this law and split it randomly into
a learning set and a test set. Figure 5 shows the distribution of the first two covariates and the
class membership indicator in the learning set. To the data of the learning set we fit feed-forward neural nets with a logistic activation function and one hidden layer with the number of hidden units \( r \) varying between 0 and 15. Figure 6 illustrates the apparent error rates observed in the learning sample and the test set based error rates. We observe that the apparent error rate decreases with increasing number of hidden units and hence suggests (incorrectly) that a neural net with many hidden units should be chosen, whereas the unbiased test set based error rate suggests a neural net with two or three hidden units and indicates that neural nets with a large number of hidden units have a higher misclassification probability. This is expected from the results of Section 3. With increasing number of hidden units we fit more and more implausible functions which move away from the true law \( f \), and hence the misclassification probability increases. On the other hand, the fitted functions resemble more and more closely the data of the learning set, and the apparent error rate decreases. In this artificial example we known the true law generating the data, thus we can compute the true misclassification probability of each rule, which is also shown in Figure 6. The test set based error rates vary around the true probabilities, which illustrates their unbiasedness.

Although such a splitting into a learning set and a test set is widely recommended, it is in general wasteful of information. One should be aware that accurate estimation of misclassification probabilities requires large samples. For example, in order to estimate a true misclassification probability of 20 per cent with an absolute standard error of 1 per cent requires a sample size of 1600. Hence it is usually desirable to use the complete sample both for construction of a rule and for estimation of misclassification probabilities. There exist several techniques for this task,
the most well-known being cross-validation. The idea of cross-validation is to split the complete data set several times into a learning set and a test set, to perform estimation of the model parameters in each learning set, and estimation of the misclassification probability in the corresponding test set, and finally to use the average of the test set based error rates as an estimate of the misclassification probability of the rule derived from estimation of the model parameters in the complete sample. Often, the sample is split $k$ times with $k$ disjoint test sets, and is referred to as $k$-fold cross-validation. If each test set contains only one data point, we speak of leave-one-out cross-validation. To illustrate the power of this approach we have used it in the above example to estimate the misclassification probability using only the learning set. The resulting error rates based on leave-one-out cross-validation are also shown in Figure 6. We can observe that these estimates vary around the true misclassification probability in a similar manner as the test set based estimates, so that they are of comparable quality.

In developing rules based on neural nets, the computation of test set based error rates is often an intermediate step in the process of finding the optimal number of hidden units. Then the neural net with the smallest test set based error rate is selected as the new rule. A common mistake is to report the minimal test set based error rate, that is, the rate observed for the selected rule, as an estimate for the misclassification probability in classifying future objects using the selected rule. However, this is a biased estimate which tends to underestimate the true misclassification
probability. A closer look at Figure 6 may illustrate the reason; the test set based error rates vary around the true misclassification probabilities and by selecting the rule with the minimal rate – here the rule based on a 5-2-1 or 5-3-1 network – we have a high probability of selecting a rate smaller than the corresponding true rate. Because a single example may not be convincing, we performed a small simulation study where we generated 500 pairs of learning and test sets of size 200 according to the above law. We always fitted as above the 16 different neural nets and selected that with minimal test set based error rate. On average we observe a true misclassification probability of 0.233, but a test set based error rate of only 0.211, that is, an average difference of 0.022. Looking at the ratio between the test set based error rate of the selected rule and the true misclassification probability of the selected rule, we observe on average an underestimation of 9 per cent. Moreover, in 75 per cent of the repetitions the test set based error rate of the selected rule was smaller than the true misclassification probability. If we consider the same problem with a smaller sample size of 100 for learning and test set (and many examples of the use of neural nets in medical applications are even smaller, see the next section), then the results are even worse; we observe an average difference of 0.036 and an average relative underestimation of 13 per cent.

A general strategy to avoid this underestimation is to split the available sample into three subsamples: learning, validation, and test set [5,22]. The learning set is used to estimate the parameters in models with a varying number of hidden units. The validation set is used for an unbiased estimation of the misclassification probability of the rule based on the parameter estimates obtained in the learning set for each model. Then the rule with the smallest error rate is selected, and finally, the test set is used to obtain an unbiased estimate of the misclassification probability of the selected rule. This is a nice paradigm, but even more wasteful of information. Again cross-validation allows to make use of all data. However, it requires a very computer time intensive nested loop. In each subset with one observation left out we have to repeat the complete process of selecting a rule, that is, for each considered model we have to estimate first the parameters and then the error rate by cross-validation. From this point of view alternative methods to allow a direct selection among different models are helpful. This can be done by adding some penalty to the minimal Kullback–Leibler distance achieved in fitting the single models, where the penalty depends on the number of parameters of each model. The most prominent measures following this idea are Akaike's information criterion [49], and its analogue, the network information criterion [50]. Using such criteria, we need cross-validation only to assess the misclassification probability of the resulting rule.

In addition to the misclassification probability, other measures for the accuracy are possible. Especially in diagnostic studies one should report additionally the sensitivity and specificity of a new rule or the whole receiver operating characteristic (ROC) curve [51]. If – as it is often the case in diagnostic studies – the sample is not drawn from one population, but separately from two populations of diseased and healthy subjects, these are the only meaningful measures, as the misclassification probability depends on the prevalence, that is, a chance of selecting a diseased or healthy subject which is arbitrary.

6. FREQUENTLY MADE MISTAKES IN APPLICATIONS OF ANNS

We reviewed the medical literature between 1991 and 1995 to examine applications of feed-forward neural networks for prognostic and diagnostic classification in oncological studies.
A Medline search conducted in May 1996 yielded 173 articles in English or German concerning the use of artificial neural networks and artificial intelligence in oncology. After reading the available abstracts, 43 articles, summarized in Table I, remained in which feed-forward neural nets were used for classification, some of which considered statistical methods as alternatives. The topics of these articles were diagnosis of cancer (23 articles), automatic tumour grading (6 articles) and prognosis (10 articles), and a further 3 reviews and 2 commentaries. Most of the excluded articles dealt with the application of artificial intelligence in image analysis and the use of expert systems for classification. We reviewed the methodological quality of these publications and identified some common mistakes which are indicated in Table I and are now described in detail.

6.1. Mistakes in estimation of misclassification probabilities

6.1.1. Biased estimation. The dramatic underestimation of the apparent error rate seems to be well-known. In none of the articles was the apparent error rate used as an estimate of the misclassification probability. For estimation, either an independent test set or cross-validation or another resampling method was always used. However, in most cases, the network with the smallest test set based error rate or the smallest cross-validation error rate was selected and this minimal error rate was regarded as a valid estimate of the true misclassification probability, which is not true, as shown in the last section. There are only a few articles following the paradigm of a split into learning, validation and test sets ensuring unbiased error rates.

6.1.2. Inefficient estimation. In many articles the size of the test set was very small; in six articles it contains less than 20 observations. Consequently, the estimates of the misclassification probabilities are highly unstable, but this problem is totally ignored. In general, all applications using a split sample approach could have been improved by using cross-validation techniques.

6.2. Fitting of implausible functions

In Section 3 we have demonstrated that due to overfitting the available data fitting neural nets with many hidden units results in implausible functions to describe the probability of class membership; there exists no generally valid limit where overfitting begins, but a key number is the ratio between the number of observations and the number of parameters in a model. In the example of Figure 3 with 39 observations we can observe that we have distinct overfitting for a 1-5-1 network with 16 parameters, and already some overfitting for a 1-3-1 network with 10 parameters. This agrees with traditional rules of thumb in the statistical literature, requiring 5 or 10 observations for each parameter to be estimated, which are also considered in the literature on neural networks [52]. If we assume that overfitting occurs at least if the ratio between the number of observations and the number of parameters is smaller than two, then in 23 of the 43 articles we are concerned that the resulting rule suffers from overfitting, that is, it is based on a highly implausible function. The use of weight decay was not mentioned in any of these articles.

6.3. Incorrectly describing the complexity of a network

The above rules of thumb are also used in the literature on neural networks. In the selected applications several authors tried to overcome a possible criticism with respect to overfitting by reporting the ratio between the number of observations and the complexity of the network. However, to measure the complexity they only use the number of input units, which is usually
Table I. Applications of feed-forward neural networks for diagnostic and prognostic classification in oncological studies.

<table>
<thead>
<tr>
<th>Year</th>
<th>First author</th>
<th>Study type*</th>
<th>Number of individuals</th>
<th>Network</th>
<th>Number of weights</th>
<th>Common mistakes†</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Training</td>
<td>Validation</td>
<td>Test</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1991</td>
<td>Dawson [53]</td>
<td>g</td>
<td>12</td>
<td>31</td>
<td>17-4-4</td>
<td>2, 5</td>
</tr>
<tr>
<td></td>
<td>Macin [54]</td>
<td>d</td>
<td>52</td>
<td>leave-one-out</td>
<td>21-10-3</td>
<td>253</td>
</tr>
<tr>
<td></td>
<td>Ostrem [55]</td>
<td>g</td>
<td>87</td>
<td>31</td>
<td>6-12-9-3</td>
<td>231</td>
</tr>
<tr>
<td></td>
<td>Piranno [56]</td>
<td>d</td>
<td>110</td>
<td>44</td>
<td>20-40-55</td>
<td>3095</td>
</tr>
<tr>
<td>1992</td>
<td>Astion [57]</td>
<td>d</td>
<td>57</td>
<td>20</td>
<td>9-15-2</td>
<td>182</td>
</tr>
<tr>
<td></td>
<td>Cicchetti [58]</td>
<td>r</td>
<td>Commentary</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Goldberg [59]</td>
<td>p</td>
<td>200</td>
<td>leave-one-out</td>
<td>3-2-1</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>McGuire [45]</td>
<td>p</td>
<td>133</td>
<td>66</td>
<td>7-6-2</td>
<td>62</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8-7-2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Nafe [60]</td>
<td>d</td>
<td>58</td>
<td>leave-one-out</td>
<td>40-14-5</td>
<td>649</td>
</tr>
<tr>
<td></td>
<td>O’Leary [61]</td>
<td>d</td>
<td>36</td>
<td>19</td>
<td>2-4-1</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>Ravdin [38]</td>
<td>p</td>
<td>500</td>
<td>420</td>
<td>8-6-1</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>Ravdin [39]</td>
<td>p</td>
<td>508</td>
<td>500</td>
<td>8-4-1</td>
<td>41</td>
</tr>
<tr>
<td>1993</td>
<td>Kappen [36]</td>
<td>p</td>
<td>269</td>
<td>10% of the data</td>
<td>17-0-1</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Schweiger [62]</td>
<td>d</td>
<td>90</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7-15-5</td>
<td>200</td>
</tr>
<tr>
<td>1994</td>
<td>Becker [63]</td>
<td>d</td>
<td>36</td>
<td>19</td>
<td>2-4-1</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>Bugliosi [64]</td>
<td>p</td>
<td>153</td>
<td>19</td>
<td>19-5-?</td>
<td>19-5-?</td>
</tr>
<tr>
<td></td>
<td>Burke [33]</td>
<td>p</td>
<td>3500</td>
<td>3500</td>
<td>3-9-1</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>Burke [65]</td>
<td>r</td>
<td>Review</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Clark [66]</td>
<td>r</td>
<td>Review</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ercal [67]</td>
<td>d</td>
<td>216</td>
<td>40%, 60%, 80%</td>
<td>14-7-1</td>
<td>113</td>
</tr>
<tr>
<td></td>
<td>Erler [68]</td>
<td>d</td>
<td>45</td>
<td>45</td>
<td>5-3-1</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>Floyd [69]</td>
<td>d</td>
<td>260</td>
<td>leave-one-out</td>
<td>8-16-1</td>
<td>161</td>
</tr>
<tr>
<td></td>
<td>Giger [70]</td>
<td>d</td>
<td>53</td>
<td>leave-one-out</td>
<td>2-4-1</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>Kolles [71]</td>
<td>g</td>
<td>?</td>
<td>?</td>
<td>?-?-?</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Macin [72]</td>
<td>d</td>
<td>72</td>
<td>leave-four-out</td>
<td>35-35-35-5</td>
<td>2700</td>
</tr>
<tr>
<td></td>
<td>Reinus [73]</td>
<td>d</td>
<td>709</td>
<td>4-fold crossval.</td>
<td>95-0-45</td>
<td>4320</td>
</tr>
<tr>
<td>1994</td>
<td>Rogers [74]</td>
<td>r</td>
<td>Review</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Snow [75]</td>
<td>p</td>
<td>1578</td>
<td>209</td>
<td>14-50-1</td>
<td>801</td>
</tr>
<tr>
<td></td>
<td>Wilding [76]</td>
<td>p</td>
<td>104</td>
<td>5-fold crossval.</td>
<td>12-5-1</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>Wu [77]</td>
<td>d</td>
<td>727</td>
<td>leave-one-out</td>
<td>9-8-1</td>
<td>89</td>
</tr>
<tr>
<td>1995</td>
<td>Attikouzel [78]</td>
<td>p</td>
<td>65</td>
<td>248</td>
<td>6-18-1</td>
<td>145</td>
</tr>
<tr>
<td></td>
<td>Baker [79]</td>
<td>d</td>
<td>477</td>
<td>952</td>
<td>4-12-1</td>
<td>73</td>
</tr>
<tr>
<td></td>
<td>Christy [80]</td>
<td>g</td>
<td>52</td>
<td>29</td>
<td>10-10-2</td>
<td>132</td>
</tr>
<tr>
<td></td>
<td>Doyle [81]</td>
<td>p</td>
<td>528</td>
<td>5-fold crossval.</td>
<td>33-48-30-1</td>
<td>3133</td>
</tr>
<tr>
<td></td>
<td>Fogel [82]</td>
<td>d</td>
<td>400</td>
<td>283</td>
<td>9-2-1</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>Gurney [83]</td>
<td>d</td>
<td>318</td>
<td>271 bootstrap.</td>
<td>7-3-1</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>Hamamoto [84]</td>
<td>p</td>
<td>54</td>
<td>11</td>
<td>9-14-1</td>
<td>155</td>
</tr>
<tr>
<td></td>
<td>Kolles [85]</td>
<td>g</td>
<td>226</td>
<td>679</td>
<td>4-30-10-3</td>
<td>493</td>
</tr>
<tr>
<td></td>
<td>Moul [86]</td>
<td>g</td>
<td>93</td>
<td>9-fold crossval.</td>
<td>4-1-1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Niederberger [87]</td>
<td>r</td>
<td>Commentary</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Simpson [88]</td>
<td>d</td>
<td>64</td>
<td>27</td>
<td>?-?-?</td>
<td>1, 4, 5, 6</td>
</tr>
<tr>
<td></td>
<td>Strotzer [89]</td>
<td>d</td>
<td>115</td>
<td>115</td>
<td>28-7-1</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>Truong [90]</td>
<td>d</td>
<td>56</td>
<td>56</td>
<td>7-10-1</td>
<td>91</td>
</tr>
</tbody>
</table>

* d, diagnosis/differential diagnosis; g, automatic grading/cytology; p, prognosis/survival; r, review/commentary
† 1, mistakes in estimation of misclassification probabilities; 2, fitting of implausible functions; 3, incorrectly describing the complexity of a network; 4, no information on complexity of the network; 5, use of inadequate statistical competitors; 6, insufficient comparison with statistical method; 7, naive application of artificial neural networks to survival data.
much smaller than the number of parameters to describe the network. Hence the reported ratios are much too large and the danger of overfitting is underestimated.

6.4. No information on complexity of the network

In order to enable the reader to judge the danger of an overfitted rule using some rule of thumb, it is necessary to know the number of hidden layers and hidden units to calculate the number of fitted weights. However, some publications even omit this basic information.

6.5. Use of inadequate statistical competitors

Some authors compared the performance of a feed-forward neural network with that of a statistical competitor, in all but one of these applications linear or quadratic discriminant analysis. All results reported were in favour of the artificial neural network. This is not surprising since feed-forward neural networks are highly flexible classification tools and ‘may yield better classification results than DFs (discriminant functions) since they form complex, non-linear associations between input data and output results’ [68]. An adequate and fair comparison of the performance of feed-forward neural networks and statistical methods must be based on statistical tools of similar flexibility like nearest-neighbour methods [91], generalized additive models [12], CART [92] or logistic regression models with quadratic terms and multiplicative interaction terms. Such comparisons were made by Ripley [7] and Vach et al. [93] for example, and demonstrated that the discriminative power of neural networks and flexible statistical methods are similar.

6.6. Insufficient comparison with statistical method

Aston and Wilding [57] concluded that ‘results from neural networks compare favorably with those obtained for the same sets of patients as analyzed by QDFA (quadratic discriminant function analysis).’ This conclusion was solely based on four (feed-forward neural network) and five (quadratic discriminant analysis) misclassified observations for a test set of 20 observations, that is, highly unstable estimates of the misclassification probabilities. No attempt was made to prove the significance of the difference between the observed misclassification rates, which is necessary to conclude that one rule outperforms the other. This testing can be done by use of the McNemar test, for example. The performance of a feed-forward neural network and a statistical competitor was evaluated by simply inspecting the estimates of the misclassification probability in several applications.

6.7. Naive application of artificial neural networks to survival data

All applications of feed-forward neural networks to survival data [33,36,38,39] suffer from the deficiencies mentioned in Section 4. The approach used in Kappen and Neijt [36] does not guarantee monotonicity of the estimated survival curves. Burke [33] omitted the censored cases, which is a serious source of bias. Ravdin and Clark [38] and Ravdin et al. [39] used the number of the time interval as additional input unit, hence the estimated survival probabilities do not depend on the length of the time intervals. These obvious deficiencies show that there is some danger that the fruitful development of statistical methodology for survival data during the last three decades may be wasted with these naive applications of neural networks to such data.
7. CONCLUSIONS

We have seen that most applications of artificial neural networks for prognosis and diagnosis in oncology during the last years suffer from methodological deficiencies. In many applications some overfitting probably occurs, resulting in biologically implausible functions to describe the class membership probability. In many applications the reported error rate for the selected neural network underestimates the true misclassification probability. Moreover, we often observed a fundamental misunderstanding of the mathematics of neural networks and basic statistical principles. Examples are the use of two output units in the case of two classes, which gives the same result as using only one output unit, the fitting of non-monotone survival curves, misunderstanding of sensitivity and specificity measures or error rates based on very small samples, ignoring the instability of such estimates. One might hope that review papers and textbooks on artificial neural networks available for any potential user (for example, Cheng and Titterington [6], Hecht-Nielsen [94] and Ripley [7]), which include warnings against the common mistakes we mentioned, may lead to a better understanding and use of artificial neural networks. However, we could not observe any improvement in the quality of applications of artificial neural networks in oncological studies over time (see Table I). In revising this paper, we picked out two recent articles on the application of artificial neural networks in oncology that appeared in prominent journals [95, 96]. In these two articles several of the commonly made mistakes appeared again, giving rise to the conjecture that the situation has not improved substantially after our literature review.

Incidentally, we came across a paper by Peter Lachenbruch entitled ‘Some misuses of discriminant analysis’ [97] where he identified frequently made mistakes of similar type. This paper was written in 1977, at a time where statistical packages like BMDP, SAS, or SPSS became widely available to non-specialists. This seems to have been a similar situation to today where numerous user-friendly neural network packages are available which can in principle be used by everyone who is able to click the mouse button on a personal computer.

The application of artificial neural networks in biomedical applications is often accompanied by grossly overstated claims, praising neural networks as the ultimate solution to the problem of diagnosis and prognosis. For example, neural networks ‘ability to learn … make them formidable tools in the fight against cancer’ [33] and ‘neural computation may be as beneficial to medicine and urology in the twenty-first century as molecular biology has been in the twentieth’ [87].

In our opinion, there is no evidence that artificial neural networks have provided real progress in the field of diagnosis and prognosis in oncology. We have tried to demonstrate that feed-forward neural networks are nothing more than regression models like logistic regression models, the only differences being that feed-forward neural networks (with hidden layers) provide a larger class of regression functions. This is often referred to as the greater flexibility of neural networks. However, greater flexibility is only of value if the true regression function is far away from that of a linear logistic model. Small deviations from a linear logistic model do not matter, because due to the small sample sizes of a few hundred typical in oncological applications, such a difference may be small relative to random errors. Large deviations, especially functions with many jumps, are not very plausible, because biological relationships tend to be smooth. Hence one cannot expect that the greater flexibility of neural networks helps them to outperform logistic models, especially if the latter are combined with careful model building, allowing use of quadratic or higher interaction terms, for example. In a recent publication, Ennis et al. [98] compared the results of a logistic regression approach [99] with that of artificial neural networks and several flexible statistical modelling approaches using data from a large clinical trial in cardiology [100].
They concluded that ‘with such a large data set none of the adaptive non-linear methods that we tried could outperform the logistic regression model of Lee et al.’. Similar experiences have been encountered in other applications in cardiology [101, 102]. Additionally, in contrast to neural network such logistic models still allow one to describe and evaluate the influence of the single covariates. This information is often of equal importance as the decision rule, as it may allow a better understanding of the underlying process.

Nevertheless, how can we explain that the comparison with statistical procedures reported in some of the selected articles almost always favour neural networks? In our opinion, this may at least partially be due to some methodological deficiencies. Usually many different neural networks are fitted on the same data set. Without using an independent test set, the minimal error rate observed within the class of neural nets underestimates the true misclassification probability. If, on the other hand, only one or two statistical procedures are used for such a comparison, corresponding error rates will not be biased to such an extent. Moreover, it seems absolutely unfair to allow an intensive model selection procedure for neural networks but not for their statistical competitors. Indeed, investigations comparing neural networks with flexible statistical methods which also allow some model building do not indicate great differences.

One should also be aware that the results favouring neural nets are based on a comparison with statistical procedures. To prove the significance of neural nets to oncology, one should at least find one application where neural nets succeed in finding a new rule proven to be better than an existing one in the literature, depending on the same covariates, which cannot be developed by statistical procedures. All applications listed in Table I avoid a comparison with an existing prognostic index or a diagnostic rule, and hence are inadequate to demonstrate the value of neural networks for medical research in oncology.

REFERENCES


