

Neural Networks as Statistical Methods in Survival Analysis

B.D. Ripley
Department of Statistics, University of Oxford
and

R.M. Ripley
Department of Engineering Science, University of Oxford

To appear in

Artificial Neural Networks: Prospects for Medicine

edited by R. Dybowski and V. Gant, Landes Biosciences Publishers

Neural networks are increasingly being seen as an addition to the statistics toolkit which should be considered alongside both classical and modern statistical methods. Reviews in this light have been given by one of us^{1,2,3,4,5} and Cheng and Titterton,⁶ and it is a point of view which is being widely accepted by the mainstream neural networks community. There are now many texts^{5,7,8,9} covering the wide range of neural networks methods; we concentrate here on methods which we see as most appropriate generally in medicine, and in particular on methods for survival data which have not to our knowledge been reviewed in depth (although Schwarzer et al.¹⁰ review a large number of applications in oncology). In particular, we point out the many different ways classification networks have been used for survival data, as well as their many flaws.

Most applications of neural networks to medicine are classification problems; that is the task is on the basis of the measured *features* to assign the patient (or biopsy or EEG or . . .) to one of a small set of classes. Baxt¹¹ gives a table of applications of neural networks in clinical medicine which are almost all of this form, including those in laboratories.¹² Classification problems include diagnosis, some prognosis problems ('will she relapse within the next three years?'), establishing depths of anaesthesia¹³ and classifying sleep state.¹⁴ Other prognosis problems are sometimes converted to a classification problem with an ordered series of categories, for example time to relapse as 0-1, 1-2, 2-4 or 4 or more years¹⁵ and prognosis after head injury.^{16,17,18} We discuss neural networks for classification and their main competitors in section 1.

Regression problems are less common in medicine, especially those which would require sophisticated non-linear methods such as neural networks. We can envisage them being used for some calibration tasks in the laboratory, but a simpler example is to predict time to death of a patient with advanced breast cancer. As methods for regression can often be applied in a clever or modified way to solve classification or survival problems, we consider them in section 2. The general idea is to replace a linear function by a neural network, which can be done within many areas of statistics.

Most prognosis problems have the characteristic that for some patients in the study set the outcome has not yet happened (or they have been lost to follow-up or died from an unrelated cause). This is known as *censoring* and has generated much statistical interest^{19,20,21,22} over the last three decades. Researchers have begun to consider how neural networks could be used within this framework, and we review this work and add some suggestions in section 3.

One important observation is that neural networks provide 'black box' methods; they may be very good at predicting outcomes but are not able to provide explanations of, say, the diagnosis or prognosis. Some of the other modern methods are able to provide explanations, and one promising idea is to fit these to the predictions of the neural network and come up with an explanation. Neural networks also lack another of the characteristics of expert systems, the ability to incorporate (easily; there is some work on 'hints'²³) qualitative information provided by domain experts.

Neural networks are powerful, and like powerful cars are difficult to drive well. For many users the power will be an embarrassment, and they may do better to use the simpler tools from modern

statistics. Because of the ‘hype’ surrounding neural networks many expensive programs have been produced which have had much more effort (and understanding) devoted to the user interface than to the algorithms used. In section 4 we point out a few of the pitfalls, but would-be users are advised to read one of the better books on the subject (or to consult an expert statistician). The statistical view has pointed out many ways to use neural networks better, but unfortunately these are still only very rarely implemented. We used the S-PLUS²⁴ statistical environment on both a PC and a Unix workstation to compute the examples, but the code used to fit neural networks was written by ourselves. (The basic code is freely available as part of the on-line material for reference 25.)

Examples

We use two cancer datasets to illustrate some of our points; note that their use here is pure illustrative and is not intended as an analysis of those sets of data. The first is on survival in months (up to 18 years, but with a median of 23 months) from advanced breast cancer, supplied by Dr J.-P. Nakache. There are 981 patients and 12 explanatory features all of which are categorical. We randomly divided this into a test set of size 500 and a training set of size 481, and assessed the methods on predictions of survival for 24 months; only 3% of the patients did not have complete follow-up to that time.

The second dataset is of 205 patients with malignant melanoma following a radical operation, and has five explanatory features. This is taken from reference 21; it is the same dataset which was analysed (with additional explanatory variables) in reference 26. Figure 1 shows that there appears to be long-term survival (from melanoma) for 65% of patients, so the survival distribution does not follow any of the standard distributions. Only 57 of the patients died from the melanoma during the study. We assessed methods on their ability to predict survival to 2500 days, by which point 86 of the patients had incomplete follow-up; our analysis shows that we expect 82 of these to have survived for 2500 days.

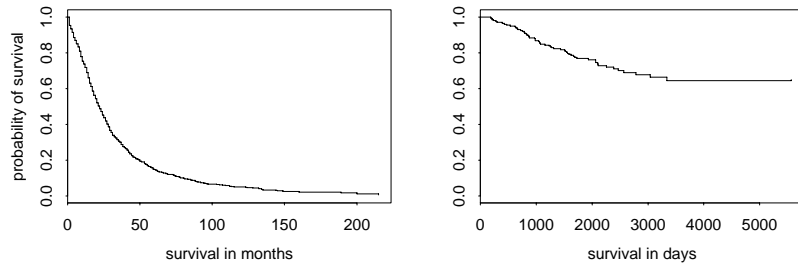


Figure 1: Plots of the Kaplan-Meier estimates of survival curves for the full (left) breast cancer and (right) melanoma datasets.

1 Classification

Suppose for the moment that we wish to classify a patient into one of two classes (for example, survival for five years or not); for many purposes it will be more helpful to know the predicted probability of survival. A simple but much neglected method is logistic regression or discrimination,⁵ which is specified by

$$P(\text{class } 2 \mid x) = \frac{e^\eta}{1 + e^\eta}, \quad \eta = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p$$

$$P(\text{class } 1 \mid x) = 1 - P(\text{class } 2 \mid x) = \frac{1}{1 + e^\eta}$$

$$\frac{P(\text{class } 2 \mid x)}{P(\text{class } 1 \mid x)} = e^\eta$$

so the explanatory variables linearly control the log-odds η in favour of class 2 (survival). The parameters β are chosen by maximum likelihood, that is by maximizing the log-likelihood

$$L = \sum_i \log P(\text{class}_i \mid x_i) \quad (1)$$

the sum being over patients. Then given the features x on a future patient we will be able to predict $P(\text{class } 2 \mid x)$, her probability of survival.

There have been many non-linear extensions of logistic regression. There are several variants of *generalized additive models*^{27,28,29} in which

$$\eta = \sum g_i(x_i)$$

where smooth functions g_i of one (or perhaps two) of the features are chosen as part of the estimation procedure, and classification trees^{30,5} in which the patients are divided into groups with a common η for each group.

The extension of logistic regression to neural networks is straightforward; we take η to be the (linear) output of a neural network with inputs x and write $\eta = g(x; \theta)$ where the parameters θ are known as ‘weights’ in the neural network literature. (Note that we can also regard this as a neural network with a single logistic output unit giving $P(\text{class } 2 \mid x)$, but that is rather coincidental.) Fitting the neural network by maximum likelihood is known as ‘entropy’ fitting in that literature and is definitely not common (and supported by amazingly few packages). It is more common to use the regression methods we discuss in section 2, which may be adequate for predicting the class (survival or death) but will be less good for predicting probabilities.

The extension to $k > 2$ classes is even less well known, although it has a long history. The idea is to take the log-odds of each class relative to one class, so the model becomes

$$\frac{P(\text{class } j \mid x)}{P(\text{class } 1 \mid x)} = e^{\eta_j}, \quad j = 2, \dots, k$$

and so

$$P(\text{class } j \mid x) = \frac{e^{\eta_j}}{\sum_{c=1}^k e^{\eta_c}}, \quad \eta_1 \equiv 0 \quad (2)$$

With $\eta_j = \beta_j^T x$ this is known as multiple logistic regression.⁵ The parameters (β_j) are fitted by maximizing the log-likelihood L given in (1). There have been surprisingly few non-linear extensions in the statistics literature; there is some recent work on additive multiple logistic regression called *POLY-CLASS*³¹ models. The extension to neural networks is easy; use (2) with (η_1, \dots, η_k) the k (linear) outputs of a neural network. (Only $k - 1$ outputs are needed, but for symmetry we do not insist that $\eta_1 = 0$.) Bridle^{32,33} gave this the pretentious title of *softmax*. Once again, softmax networks are not implemented in most neural network packages; rather they provide networks with k logistic outputs, which amounts to using

$$P(\text{class } j \mid x) = \frac{e^{\eta_j}}{1 + e^{\eta_j}}, \quad j = 1, \dots, k$$

This is an appropriate model for diagnosis where a patient might have none, one or more out of k diseases, but not for general classification problems.

Classification for prognosis problems

It is surprising how often classification networks have been applied to prognosis problems, especially as it would seem that the methods we consider in section 3 would often be more appropriate. (This is probably due to the ready availability of software for classification networks.) There are many variants. We usually have to take *censoring* into account, that is that follow-up on some patients may end before the event (which we describe as ‘death’).

1. The simplest idea^{34,35,36} considers survival for some fixed number of months or years, and ignores patients censored before that time, thereby giving a standard two-class classification problem. Omitting censored patients may bias the result, however. Imagine a study of survival for five years after an operation where most deaths occur in the post-operative phase, all patients have been followed up for three years but few for the full five years. Then the censored patients are very likely to have survived for five years, and the estimates of the survival probabilities will be biased downwards. This bias may not be important in explaining the variations in survival from the explanatory features, but these studies are concerned with predicting not explaining.

Ravdin and Clark³⁷ give an example of this effect: in their study 268 patients had known follow-up for 60 months, of whom 213 had died although the Kaplan-Meier estimate of the survival probability was 50%. We can also see this in our melanoma example. Of those patients with complete follow up to 10 years, 23 out of 80 survived, yet the Kaplan-Meier estimate of survival for this time is 64.5%.

2. A refinement is to divide the survival time into one of a set of non-overlapping intervals, giving an ordered series of k classes. (For definiteness let us take the classes ‘death in year 1’, ‘death in year 2’, ‘death in year 3’ and ‘survive 3 or more years’.) This can be done in a number of ways. Perhaps the most natural is to use a proportional odds model³⁸ for the ordered outcomes. It is much more common to ignore the ordering of the classes, and to use a k -class classification network.^{39,40,15} The perceived difficulty is how to handle censoring: sometimes all censored patients are ignored (but this causes a bias in the predictions). The remedy is in fact theoretically easy: for example the contribution to the log-likelihood L for a patient who was lost to follow up after 2 years is

$$\log\{P(\text{death in year 3} \mid x) + P(\text{survive 3 or more years} \mid x)\}$$

This does however need modifications to the software, so standard methods for fitting classification networks cannot be used. If this is done there is only a small bias, due to the fact that censored patients will have survived some of the interval in which they were lost to follow-up.

These methods produce a crude estimate of the survivor curve $S(t) = P(\text{alive at time } t)$ by taking one minus the cumulative probabilities across classes. If a prediction of prognosis is required we clearly should not take the class with the largest predicted probability (especially if the intervals are of unequal length); a good choice would be the interval over which the cumulative probability of death moves from below 50% to above 50%.

3. Other authors use k separate networks. This can be done in one of two ways: in our example we could use networks for either (a) the original four classes⁴¹ or (b) for the three classes^{42,43,44} ‘death in year 1’, ‘death in year 1 or 2’ and ‘death in years 1, 2 or 3’. In either case we can train each network on those patients with follow-up past the end of the interval, so that later networks are trained on less data, and once again there are problems of bias.

It is easy for networks trained with option (b) to give inconsistent answers, for example to give a higher predicted probability for ‘death in year 1 or 2’ than for ‘death in years 1, 2 or 3’. This was reported by Ohno-Machado and Musen⁴⁴, who try to circumvent this by using the output of one network (say ‘death in year 1 or 2’) as an input to the others. However, such difficulties are indicative of a wrong formulation of the problem. (Surprisingly, that paper does not mention the more satisfactory approach⁴⁰ of using a k -output network used on the same dataset by one of its authors!)

Lapuerta et al.³⁹ used a network with four outputs corresponding to death in one of three 40-month periods or survival for ten years for their final predictions. However, during training they coped with censored data by imputing a death period for those patients lost to follow-up. This was done by training separate networks for death in periods 2 and 3. The features on a patient

lost to follow-up during period 1 were input to the period 2 network; if that predicted death, death in period 2 was assigned but if not the period 3 network was used to impute either death in period 3 or survival for ten years.

Ravdin et al.⁴⁵ have a variation on theme (b), in which they combine the k separate networks into one network with an additional input, the number of years for which survival is to be predicted. The training set repeats each patient for all the numbers of years for which survival or death is known. Ravdin and Clark³⁷ extend this approach by attempting to ameliorate the problems of bias by randomly selecting a proportion of the deaths to match the proportion given by a classical Kaplan-Meier estimate of the survival curve. (This is not an exact procedure; if it is to be used it would be better to weight cases than to randomly choose them.)

4. Another alternative¹⁹ is to model the conditional probabilities

$$P(\text{die in } i\text{th interval} \mid \text{survive first } i-1 \text{ intervals}, x) = g(\eta_i)$$

where g is usually the logistic function $e^x/(1 + e^x)$. Then a patient dying in the i th interval contributes $\log\{g(\eta_i)[1 - g(\eta_{i-1})] \cdots [1 - g(\eta_1)]\}$ to the log-likelihood, and a patient lost to follow up in that interval $\log\{[1 - g(\eta_{i-1})] \cdots [1 - g(\eta_1)]\}$, and from this the log-likelihood L can be computed. The ‘scores’ η_1, \dots, η_k are given by the output of a neural network with k linear outputs. (This model can be regarded as a ‘life-table’ or discrete-time survival model,²⁰ and is sketched in those terms by Liestøl et al.²⁶ It is sometimes known as a ‘chain-binomial’ model.)

It is possible^{46,47} to fit this model using standard neural-network software (although the predictions do have to be post-processed.) We can expand the contribution to the log likelihood as a sum of $\log g(\eta_i)$ or $\log[1 - g(\eta_i)]$ over the intervals for which that patient is at risk. This is computed by having an additional input to the neural network specifying the time interval i for which $g(\eta_i)$ is required, and entering each patient into the training set for each time interval until death or the end of follow-up. Thus the training set (both inputs and outputs) is similar to that used by Ravdin et al., but patients are not entered after death and the fitted network is used in a different way. Note that although this technique is possible, special-purpose software will be substantially more efficient.

This method also has only a small bias due to censoring; it is equivalent to approach 2 but uses a different parametrization of the survival probabilities.

It may be helpful to re-state the censoring problem in mathematical terms. Suppose we have $k+1$ time intervals, $[0 = t_0, t_1), [t_1, t_2), \dots, [t_{k-1}, t_k), [t_k, \infty)$, and let $s_i = S(t_i)$ be the probability that a patient survives to time t_i , and suppose we are particularly interested in s_k . Approaches 1 and 3 estimate s_k directly. Approach 2 estimates $p_i = P(t_{i-1} \leq T < t_i)$ and then $s_k = p_{k+1} = 1 - p_1 - \dots - p_k$. Approach 4 estimates $g_i = P(t_{i-1} \leq T < t_i \mid T > t_{i-1})$, and then $s_k = (1 - g_1) \cdots (1 - g_k)$. Approaches 2 and 4 are able to (approximately) adjust for censoring since a patient lost to follow-up in the interval $[t_{i-1}, t_i)$ is counted as a survivor in estimating p_1, \dots, p_{i-1} or g_1, \dots, g_{i-1} rather than being ignored.

Unfortunately, the only methods that deal correctly with censoring use a different log-likelihood from that used in standard packages, and hence need software modifications or use the software inefficiently. The approaches of Biganzoli et al.⁴⁷ and Lapuerta et al.³⁹ are the most satisfactory of those using standard software.

2 Regression problems

Many neural network packages can only tackle regression problems; that is they are confined to fitting functions $g_j(x; \theta)$ by least squares, minimizing

$$\sum_i \sum_{j=1}^k [y_{ij} - g_j(x_i; \theta)]^2$$

the first sum being over patients. This corresponds to $k \geq 1$ non-linear regressions on the explanatory variables x . The most common usage is a neural network with a single linear output (for calibration in pyrolysis mass spectrometry, for example) or with a logistic output for a two-class classification problem. It would seem obvious to take $y = 1$ for survival and $y = 0$ for death, but as we saw in section 1, the use of least-squares is not really appropriate and ‘fudges’ have grown up such as coding survival as $y = 0.9$ and death as $y = 0.1$. The extension to a k -class classification problem is to take $y_{ij} = 1$ for the class which occurred and $y_{ij} = 0$ for the others; then when the network is used for the prediction the class with the largest output is chosen. (Other ways to use regression methods for classification problems are discussed in chapter 4 of reference 5.)

There has been a parallel development of nonlinear regression methods in statistics. Additive models are of the form

$$g_j(x; \theta) = \alpha_j + \sum_{s=1}^p \beta_{js} g_s(x_s; \theta)$$

which allow a nonlinear transformation of each of the features. The functions g_s can be chosen nonparametrically²⁷ or by smoothing splines²⁸; some implementations such as MARS⁴⁸ also allow functions of more than one feature. Perhaps the most wide-ranging generalization of additive models is *projection pursuit regression*⁴⁹ which is an additive model in linear combinations of the features. This subsumes neural networks with a single hidden layer, but the algorithms developed in the statistical literature for fitting projection pursuit regressions are less powerful than those now known for fitting neural networks.

Classification trees have a counterpart, regression trees,³⁰ in which once again the patients are grouped and a constant value assigned to each group; the groups are found by a tree-structured set of rules.

Great ingenuity has been shown in finding ways to apply existing regression methods and software to other problems. For example, Therneau et al.⁵⁰ suggest applying regression trees to the residuals from a linear survival analysis to provide a nonlinear survival method using existing software, and this idea could equally be applied to neural networks.

3 Survival analysis

The conventional setup in survival analysis is that there is a time-to-outcome T which is measured continuously plus a censoring indicator δ which indicates whether the outcome was ‘death’ ($\delta = 1$) or the patient was lost to follow-up ($\delta = 0$). The standard statistical procedures^{20,22,25} relate the distribution of T to explanatory variables x via a *linear predictor* $\eta = \beta^T x$. For example, proportional hazards models have the hazard at time t (the rate of death at time t of those who are still alive)

$$h(t) = h_0(t)e^\eta \quad (3)$$

where $h_0(\cdot)$ is known as the baseline hazard, and an accelerated life model fits a standard distribution to $Te^{-\eta}$, so the linear predictor speeds up or slows down time for that patient. We discuss below how these models can be generalized to use neural networks.

Parametric models for survival analysis can be very useful but are often neglected. Common choices for a parametric proportional hazards model are the Weibull distribution and its special case

the exponential, and for accelerated life models the Weibull (again) and the log-logistic. However, following Cox,¹⁹ the semi-parametric proportional hazard model has become extremely popular. This assumes (3) with no assumption on the baseline hazard and η is estimated by partial or marginal likelihood methods.²⁰

Nonlinear models in survival analysis are surprisingly rare in the statistical literature. There are a few references^{51,52,53,54} suggesting additive extensions of Cox models as well as a fully local approach,⁵⁵ and a modest literature^{56,57,58,59,60} on tree-structured survival analysis.

The only previous attempt of which we are aware to apply neural networks directly to survival analysis is by Faraggi and Simon,⁶¹ applied by Mariani et al.⁶² They consider partial-likelihood estimation of model (3) with $\eta = f(x; \theta)$ the output of a neural network. We have implemented this and the parametric models mentioned earlier. We should point out that there is a much easier way to fit Cox models with η given by a neural network, which is to use an iterative idea.^{52,59} This alternates estimating the baseline cumulative hazard $H_0(t)$ by the Breslow estimator and choosing θ to maximize

$$\sum_i \left\{ \delta_i \eta_i - H_0(t_i) \exp \eta_i \right\} \quad \eta_i = f(x_i; \theta)$$

(the sum being over patients) starting with $\eta_i \equiv 0$ or with a linear fit. Normally only a couple of iterations are required. The solution is a (local) maximum of the partial likelihood.

4 Fitting neural networks

Perhaps the major cause of difficulty in fitting neural networks is the ease with which it is possible to overfit, that is to tune the neural network to the peculiarities of the examples to hand rather than to extract the salient dependencies of the whole population. In a phrase borrowed from psychology, we want to fit a network to achieve good *generalization*. Why is this an especial problem for neural networks? In using classical statistical methods we build up from simple models, perhaps first fitting a linear model and then allowing quadratic or interaction terms and at each stage testing for a significant improvement in fit. There is no analogue for neural networks, and there are results⁵ that show that with enough hidden units we can make (essentially) arbitrarily complicated models.

For good generalization we do not want to use maximum likelihood fitting (or least-squares fitting). We borrow the ideas of *regularization* from the numerical methods field, and penalize ‘rough’ functions $f(x; \theta)$. This is most conveniently done using *weight decay* in which we maximize

$$L - \lambda \sum_{\text{weights}} w_{ij}^2$$

How do we choose λ ? There are some very effective guidelines⁵ based on statistical ideas, but as with the number of hidden units it is best chosen by a validation experiment.

Not only does weight decay help to achieve good generalization, it also makes the optimization task easier and so faster. Thus it is very surprising that (yet again) it is omitted from most packages, yet most experts in the field believe that it should *always* be used. Instead, most packages use the older idea of *early stopping* with an inefficient method of optimization; this will usually work but can be one or two orders of magnitude slower and is responsible for the reputation that neural networks have of being very computationally demanding. (*None* of the application studies we reviewed used weight decay nor explained how training was stopped nor how the number of hidden units were chosen. Mariani et al.⁶² are a commendable exception which appeared whilst this paper was in preparation.)

Although a neural network can handle complicated relationships, it is likely to generalize better if the problem is simplified, so as much care in preparing the data and transforming the inputs should be used for neural networks as for conventional statistical methods.

In the vast majority of neural network fitting problems there will be multiple local optima, so if the optimization is run from a different set of initial weights, different predictions will be made.

Sometimes the differences between predictions at different local optima will be small, but by no means always. (Reference 5 has some simple examples for a medical diagnosis problem.) It is *not* a good idea to choose the best-fitting solution (that is probably the one that overfits the most); it is better to combine the predictions from the multiple solutions. The idea of averaging the probability predictions across, say, 25 fits is rather effective, and many other averaging ideas^{63,64,65,66} have been suggested.

Several studies claimed that their neural network model outperformed a Cox regression and/or clinicians, but such findings need to be examined critically. None of the studies considered using non-linear terms nor interaction terms in the Cox regression, and this would be standard practice for a statistical expert using such models. However, the basis of the comparison is flawed. Cox models are not designed to estimate the probability of survival at a fixed time (usually the end of the study); they are intended to show the dependence of the survivor curve on the explanatory features. Even when used for prediction, they are able to predict the whole survivor curves, and it is not surprising that they are less able to predict one point on that curve than methods designed to predict just that point (for example, logistic discrimination). Further, censoring biases in the test set will almost always favour the neural network models, which estimate the probability of survival to a fixed time *conditional* on the patient still being under follow-up, not the unconditional probability estimated by a survival-analysis model or being assessed by the clinicians. The only way to ensure a fair comparison on a test set is to impute an outcome to each patient whose follow-up is for less than the fixed time. We suggest that this is best done by grouping test-set patients on the basis of survival experience (perhaps using a tree-structured analysis to do the grouping), fitting a Kaplan-Meier survival curve to each group and using this to estimate the probability of survival of those patients in the group whose follow-up period is too short.

A frequent mistake is to take too small a test set; several authors have used a test set of less than 20 observations.¹⁰ However, the size of the test set is not the whole story, as there needs to be sufficient cases that survive and sufficient that die. The study of Bottaci et al.^{41,67} has gained considerable publicity, yet is based on the apparent success in predicting the death of just 7 out of 92 patients, and a higher accuracy (the headline measure used) would have been obtained by predicting survival for all the patients!

5 Examples

We tried most of the methods described here on one or both of the examples. Selecting the number of units in the neural networks and the amount of weight decay to be used was done by cross-validation,⁵ for a set of about a dozen values chosen from past experience. The measure of fit used was the *deviance*, summing minus twice the logarithms of the predicted probability of the event over all patients in the training set. (This provides a more sensitive measure of fit than the success rate, especially in the survival analysis models where the exact time of death is used.)

Breast cancer

We used a training set of size 500, and tested on a test set of size 476 (ignoring those 5 patients in the full test set whose follow-up to 24 months was incomplete). All the linear methods used selection of the input variables by AIC⁵; for all the methods using neural networks the number of hidden units and the amount of weight decay was chosen by 10-fold cross-validation within the training set. Our results are summarized in table 1. There *sensitivity* is the probability of correctly predicting death, *specificity* is the probability of correctly predicting survival, and the *accuracy* is the percentage of correct predictions.

There is almost nothing to choose between the methods, except that the Weibull survival models are slightly (but not significantly) poorer. This might have been expected as figure 1 shows that the overall survival distribution is not very close to Weibull. The regression methods were done with response the logarithm of survival time (using time directly gave very much worse results). This is formally equivalent to log-normal survival-analysis model, and further investigations showed that the

method	linear			neural net		
	specificity	sensitivity	accuracy	specificity	sensitivity	accuracy
binary classification	73	62	67	72	64	68
1-year periods	72	63	68	72	65	68
proportional odds				71	62	66
regression	66	68	67	63	71	67
proportional hazards	70	62	66	71	62	66
Weibull survival	72	58	64	72	61	66
log-logistic survival	70	66	67	68	66	67

Table 1: Results (%) for predictions on the test set of the breast-cancer example.

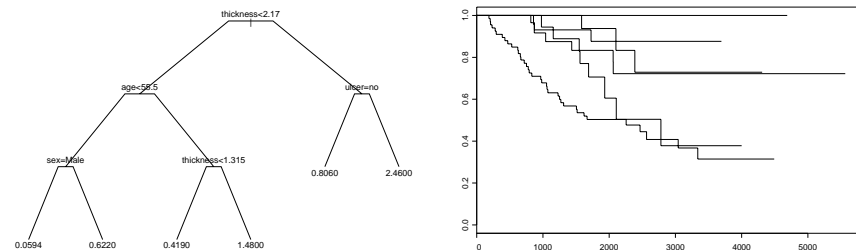


Figure 2: **Left:** Tree used to split the Melanoma data into six groups. At each node the label indicates the condition to go down the left branch, and the numbers are the hazards for the groups relative to the whole dataset. **Right:** Kaplan-Meier plots of survival in the six groups.

bias towards survival of the regression models is due to the exclusion of six cases with incomplete follow-up to 24 months (which were also excluded for the binary classifications).

Melanoma

This is a small dataset (205 patients) with heavy censoring. We used 5-fold cross-validation to assess the models: that is we randomly divided the dataset into 5 parts and for each fitted to the remaining four parts and predicted survival on the single part. Because there was heavy censoring, assessment on just those patients with complete follow-up to 2500 days would be seriously biased. We used a tree-based analysis to divide the dataset into six groups (figure 2) with homogeneous survival experience, fitted Kaplan-Meier survival curves to each group, and used these to estimate the probability that the patient would have survived from the end of observed follow up to 2500 days. (This probability was often one, and never less than 0.45.) These patients were then entered into the test set with both possible outcomes, weighted by the estimated probabilities.

The multiple-output classification problem had classes as 0–1500, 1500–2000, 2000–2500 and 2500– days, chosen by looking at the pattern of censoring times.

The results are shown in table 2. Despite the use of nested cross-validation (so that evaluating each neural network method involved $5 \times 5 \times 12$ fits) the total computation time was less than an hour. Again there are generally small differences between the methods (except for the binary classifications ignoring censoring), even though the Weibull and log-logistic distributions cannot model long-term survival as shown in figure 1. The large differences between sensitivity and specificity is not really surprising given that only about 28.2% of patients die within 2500 days. Thus we would achieve a higher accuracy than *all* of the methods by declaring all patients to survive. The underlying difficulty is that is hard to find prognostic patterns, and the dominance of survival leads to predicted probabilities of death of individual patients which are above 28% but do not reach 50%. If we consider the cost of

method	linear			neural net		
	specificity	sensitivity	accuracy	specificity	sensitivity	accuracy
binary classification	45.1	40.1	43.7	63.4	60.7	62.6
full training set	90.9	13.2	69.0	90.9	16.5	69.9
4-class	88.1	21.5	69.3	92.2	14.8	70.4
proportional odds	90.4	23.9	71.6	91.0	18.7	70.6
proportional hazards	84.3	32.8	69.8	87.6	34.3	72.6
Weibull survival	87.0	25.8	69.8	87.0	24.1	69.3
log-logistic survival	86.4	36.2	72.3	84.2	34.3	70.1

Table 2: Results (%) from 5-fold cross-validation of the melanoma example. The second row of binary classification is using the estimated probabilities as targets for the patients with incomplete follow-up to 2500 days; these patients are completely ignored in the first line.

failing to spot a death as twice that of incorrectly predicting death, a different pattern emerges shown in table 3. (With this cost pattern we predict death if the probability of survival is less than 2/3.)

method	linear			neural net		
	specificity	sensitivity	loss	specificity	sensitivity	loss
binary classification	25.1	56.6	160.5	39.6	74.7	118.1
full training set	75.5	46.5	98.0	73.5	55.3	90.8
4-class	74.3	50.5	95.0	76.2	46.6	96.8
proportional odds	74.8	46.6	98.8	76.2	51.8	90.8
proportional hazards	79.6	58.7	88.4	71.7	64.6	82.6
Weibull survival	73.5	53.5	92.8	72.4	64.6	81.6
log-logistic survival	72.1	53.5	94.8	69.0	66.4	84.5

Table 3: Results for the melanoma data with differential costs of errors. The sensitivities and specificities are percentages, whereas the losses are totals over 205 patients.

Under this cost pattern the methods from survival analysis do show a clear superiority, and within that class the non-linear methods show a substantial advantage over the linear ones. However, as this dataset is so small, only the larger differences (those between the first ‘binary classification’ line and the rest) are statistically significant when assessed by paired t -tests.

All the methods had been set up to predict probabilities of observed events, so it was easy to recompute the results for a difference pattern of costs. There are technical arguments⁵ that suggest we might have obtained (slightly) improved results by taking the cost pattern into account during training by weighting examples in the training set.

Acknowledgements

RMR was supported by an EPSRC grant during this work. Mark Mathieson supplied the code for the proportional-odds models.

References

1. Ripley BD. Statistical aspects of neural networks. In: Barndorff-Nielsen OE, Jensen JL, Kendall WS, eds., *Networks and Chaos—Statistical and Probabilistic Aspects*. London: Chapman & Hall, 1993:40–123.
2. Ripley BD. Neural networks and related methods for classification. *J R Statist Soc B* 1994; 56:409–456.

3. Ripley BD. Neural networks and flexible regression and discrimination. In: Mardia KV, ed., *Statistics and Images 2*. Abingdon: Carfax, 1994:39–57.
4. Ripley BD. Flexible non-linear approaches to classification. In: Cherkassky V, Friedman JH, Wechsler H, eds., *From Statistics to Neural Networks. Theory and Pattern Recognition Applications*. Berlin: Springer, 1994:105–126.
5. Ripley BD. *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press, 1996.
6. Cheng B, Titterton DM. Neural networks: a review from a statistical perspective. *Statistical Science* 1994; 9:2–54.
7. Hertz J, Krogh A, Palmer RG. *Introduction to the Theory of Neural Computation*. Redwood City, CA: Addison-Wesley, 1991.
8. Haykin S. *Neural Networks. A Comprehensive Foundation*. New York: Macmillan College Publishing, 1994.
9. Bishop CM. *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press, 1995.
10. Schwarzer G, Vach W, Schumacher M. On the misuses of artificial neural networks for prognostic and diagnostic classification in oncology. Tech. rep., Center for Data Analysis and Model Building, University of Freiburg, 1997.
11. Baxt WG. Application of artificial neural networks to clinical medicine. *Lancet* 1995; 346:1135–1138.
12. Dybowski R, Gant V. Artificial neural networks in pathology and medical laboratories. *Lancet* 1995; 346:1203–1207.
13. Watt RC, Sisemore CS, Kanemoto A, et al. Neural networks applied to the bispectral analysis of EEG during anesthesia. *Anesthesiology* 1995; 83:A503.
14. Pardey J, Roberts S, Tarassenko L, et al. A new approach to the analysis of the human sleep/wakefulness continuum. *J Sleep Res* 1996; 5:201–210.
15. Ripley RM, Harris AL, Tarassenko L. Neural networks for breast cancer prognosis. submitted 1997.
16. Titterton DM, Murray GD, Murray LS, et al. Comparison of discrimination techniques applied to a complex data set of head injured patients. *J R Statist Soc A* 1981; 144:145–174.
17. Lowe D, Webb A. Exploiting prior knowledge in network optimization: an illustration from medical prognosis. *Network* 1990; 1:299–323.
18. Mathieson MJ. Ordered classes and incomplete examples in classification. In: Mozer MC, Jordan MJ, Petsche T, eds., *Advances in Neural Information Processing Systems 9*. Cambridge, MA: MIT Press, 1997:550–556.
19. Cox DR. Regression models and life-tables (with discussion). *J R Statist Soc B* 1972; 34:187–220.
20. Kalbfleisch JD, Prentice RL. *The Statistical Analysis of Failure Time Data*. New York: Wiley, 1980.
21. Andersen PK, Borgan Ø, Gill RD, et al. *Statistical Models based on Counting Processes*. New York: Springer, 1993.
22. Collett D. *Modelling Survival Data in Medical Research*. London: Chapman & Hall, 1994.
23. Abu-Mostafa YS. Machines that learn from hints. *Scientific American* 1995; 272(4):64–69.
24. MathSoft Data Analysis Products Division. *S-PLUS*. Seattle, 1987–97.
25. Venables WN, Ripley BD. *Modern Applied Statistics with S-PLUS*. New York: Springer, 2nd edn., 1997.
26. Liestøl K, Andersen PK, Andersen U. Survival analysis and neural nets. *Statist Med* 1994; 13:1189–1200.
27. Hastie TJ, Tibshirani RJ. *Generalized Additive Models*. London: Chapman & Hall, 1990.
28. Wahba G. *Spline Models for Observational Data*. Philadelphia: SIAM, 1990.

29. Wahba G, Gu C, Wang Y, et al. Soft classification a.k.a. risk estimation via penalized log likelihood and smoothing spline analysis of variance. In: Wolpert DH, ed., *The Mathematics of Generalization*. Reading, MA: Addison-Wesley, 1995:331–359.
30. Breiman L, Friedman JH, Olshen RA, et al. *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks/Cole, 1984.
31. Kooperberg C, Bose S, Stone CJ. Polychotomous regression. *J Amer Statist Assoc* 1997; 92:117–127.
32. Bridle JS. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In: Fogelman Soulié F, Héroult J, eds., *Neuro-computing: Algorithms, Architectures and Applications*. Berlin: Springer, 1990:227–236.
33. Bridle JS. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In: Touretzky DS, ed., *Advances in Neural Information Processing Systems 2*. San Mateo, CA: Morgan Kaufmann, 1990:211–217.
34. Bugliosi R, Tribalto M, Avvisati G. Classification of patients affected by multiple myeloma using a neural network software. *Eur J Haematol* 1994; 52:182–183.
35. Burke HB, Rosen DB, Goodman PH. Comparing the prediction accuracy of artificial neural networks and other statistical models for breast cancer survival. In: Tesauro G, Touretzky DS, Leen TK, eds., *Advances in Neural Information Processing Systems 7*. Cambridge, MA: MIT Press, 1995:1063–1067.
36. Tarassenko L, Whitehouse R, Gasparini G, et al. Neural network prediction of relapse in breast cancer patients. *Neural Comp Appl* 1996; 4:105–113.
37. Ravdin PM, Clark GM. A practical application of neural network analysis for predicting outcome of individual breast cancer patients. *Breast Cancer Res Treat* 1992; 22:285–293.
38. Mathieson MJ. Ordinal models for neural networks. In: Refenes APN, Abu-Mostafa Y, Moody J, et al., eds., *Neural Networks in Financial Engineering*. Singapore: World Scientific, 1996:523–536.
39. Lapuerta P, Azen SP, LaBree L. Use of neural networks in predicting the risk of coronary-artery disease. *Comp Biomed Res* 1995; 28:38–52.
40. Ohno-Machado L. A comparison of Cox proportional hazards and artificial neural network models for medical prognosis. *Comput Biol Med* 1997; 27:55–65.
41. Bottaci L, Drew PJ, Hartley JE, et al. Artificial neural networks applied to outcome prediction for colorectal cancer patients in separate institutions. *Lancet* 1997; 150:469–472.
42. Kappen HJ, Neijt JP. Neural network analysis to predict treatment outcome. *Ann Oncol* 1993; 4:S31–S34.
43. Theeuwes M, Kappen B, Neijt J. Neural network analysis to predict treatment outcome in patients with ovarian cancer. In: Fogelman Soulié F, Dreyfus G, eds., *Proceedings Session 5, Medicine, International Conference on Artificial Neural Networks*. Paris, France, 1995.
44. Ohno-Machado L, Musen MA. Modular neural networks for medical prognosis: quantifying the benefits of combining neural networks for survival prediction. *Connection Science* 1997; 9:71–86.
45. Ravdin PM, Clark GM, Hilsenbeck SG, et al. A demonstration that breast cancer recurrence can be predicted by neural network analysis. *Breast Cancer Res Treat* 1992; 21:47–53.
46. Efron B. Logistic regression, survival analysis and the Kaplan-Meier curve. *J Amer Statist Assoc* 1988; 83:414–425.
47. Biganzoli E, Boracchi P, Mariani L, et al. Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach. *Statist Med* 1998.
48. Friedman JH. Multivariate adaptive regression splines (with discussion). *Ann Statist* 1991; 19:1–141.
49. Friedman JH, Stuetzle W. Projection pursuit regression. *J Amer Statist Assoc* 1981; 76:817–823.
50. Therneau TM, Grambsch PM, Fleming TR. Martingale-based residuals for survival models. *Biometrika* 1990; 77:147–160.
51. O’Sullivan F. Nonparametric estimation of relative risk using using splines and cross-validation. *SIAM J Sci Statist Comp* 1988; 9:532–542.
52. Gentleman R, Crowley J. Local full likelihood estimation for the proportional hazards model. *Biometrics* 1991; 47:1283–1296.
53. Gray RJ. Flexible methods for analyzing survival data using splines, with applications to breast cancer prognosis. *J Amer Statist Assoc* 1992; 87:942–951.
54. Kooperberg C, Stone CJ, Truong YK. Hazard regression. *J Amer Statist Assoc* 1995; 90:78–94.
55. Gray RJ. Hazard rate regression using ordinary nonparametric regression smoothers. *J Comp Graph Statist* 1996; 5:190–207.
56. Ciampi A, Chang CH, Hogg S, et al. Recursive partitioning: a versatile method for exploratory data analysis in biostatistics. In: MacNeil IB, Umphrey GJ, eds., *Biostatistics*. Dordrecht: Reidel, 1987:23–50.
57. Segal MR. Regression trees for censored data. *Biometrics* 1988; 44:35–47.
58. Davis R, Anderson J. Exponential survival trees. *Statist Med* 1989; 8:947–961.
59. LeBlanc M, Crowley J. Relative risk trees for censored survival data. *Biometrics* 1992; 48:411–425.
60. LeBlanc M, Crowley J. Survival trees by goodness of split. *J Amer Statist Assoc* 1993; 88:857–867.
61. Faraggi D, Simon R. A neural network model for survival data. *Statist Med* 1995; 14:73–82.
62. Mariani L, Coradini D, Biganzoli E, et al. Prognostic factors for metachronous contralateral breast cancer: A comparison of the linear Cox regression model and its artificial neural network extension. *Breast Cancer Res Treat* 1997; 44:167–178.
63. Wolpert DH. Stacked generalization. *Neural Networks* 1992; 5:241–259.
64. Perrone MP, Cooper LN. When networks disagree: Ensemble methods for hybrid neural networks. In: Mammone RJ, ed., *Artificial Neural Networks for Speech and Vision*. London: Chapman & Hall, 1993:126–142.
65. Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. In: *Proceedings of the Second European Conference on Computational Learning Theory*. Springer, 1995:23–37.
66. Breiman L. Bagging predictors. *Machine Learning* 1996; 24:123–140.
67. Dobson R. Program predicts cancer deaths. *The Sunday Times* 28th Sept 1997 ‘Innovations’ section, 1997.