Biometrical Journal 39 (1997) 5, 519–532 Akademie Verlag

Bayesian Neural Network Models for Censored Data

D. FARAGGI

Department of Statistics
University of Haifa
Israel

R. SIMON

Biometric Research Branch
National Cancer Institute
NIH
Bethesda
U.S.A.

E. YASKIL

Department of Statistics
University of Haifa
Israel

A. KRAMAR

Centre Regional De Lutte Contre Le Cancer
Montpellier
France

Summary

Neural networks are considered by many to be very promising tools for classification and prediction. The flexibility of the neural network models often result in over-fit. Shrinking the parameters using a penalized likelihood is often used in order to overcome such over-fit. In this paper we extend the approach proposed by FARAGGI and SIMON (1995a) to modeling censored survival data using the input-output relationship associated with a single hidden layer feed-forward neural network. Instead of estimating the neural network parameters using the method of maximum likelihood, we place normal prior distributions on the parameters and make inferences based on derived posterior distributions of the parameters. This Bayesian formulation will result in shrinking the parameters of the neural network model and will reduce the over-fit compared with the maximum likelihood estimators. We illustrate our proposed method on a simulated and a real example.

Key words: Bayesian analysis; Feed-forward neural network; Maximum likelihood; Shrinkage; Sufficiency principle.

1. Introduction

Multi-layer perceptrons are of increasing importance for problems of prediction and classification (HOLLEY and KARPLUS, 1989). Neural networks have received much attention recently by computer scientists, neurophysiologists, psychologists and engineers interested in biological nervous system organization and artificial intelligence. For example, neural networks have been used for speech recognition (SEJNOWSKI and ROSENBERG, 1987), diagnostic image analysis (DAPONTE and SHERMAN 1991), clinical diagnosis (MANN and BROWN, 1991), macromolecule sequence analysis and prediction of mechanism of action for new cancer drugs (WEINSTEIN et al., 1992).

Single hidden layer networks can approximate any continuous function as the number of hidden nodes increase (GALLANT and WHITE, 1991). This flexibility results from the presence of a large number of parameters and can result in over-fitting the data and consequent loss of generalization ability. The problem of over-fitting has been discussed by GERMAN, BIENSTOCK, and DOURSAT (1992), RIPLEY (1994), and others.

PLAUT, NOWLAN, and HINTON (1986) describe minimization of a penalized objective function as a way of avoiding over-fitting. The objective function used was the error sum of squares and the penalty consisted of a constant times the sum of squares of the estimated parameters. This method was also used by RIPLEY (1994) and the connection between it and a Bayesian model was described by BUNTINE and WEIGEND (1991) and MACKAY (1992).

We will describe the relationship to the penalized objective function approach and a Bayesian model in the context of the proportional hazard neural network model for survival data (FARAGGI and SIMON, 1995a). We shall introduce a more computationally efficient approach to a Bayesian neural network for survival data. Our proposed method is based on large sample approximation and the sufficiency principle applied to the maximum likelihood estimator. We will compare it to the maximum likelihood neural network for survival data with regard to loglikelihood and a concordant prediction index on a separate validation set not used for model fitting.

Although the Bayesian analysis in this paper is described in the context of the proportional hazards model, it can be generalized to other maximum likelihood neural network models. These can be based on other censored data models such as the accelerated failure time model (PRENTICE and KALBFLEISCH, 1979) or the Buckely-James model (BUCKELY and JAMES, 1979). Alternatively, for uncensored data, classification neural networks models (FARAGGI and SIMON, 1995b) can serve as the basis for the Bayesian analysis presented here.

2. A Bayesian Proportional hazards neural network model

2.1 *The neural network*

Neural networks, in general, consist of an input layer, one or more hidden layers and an output layer. In this paper we consider networks with one layer of "hid-

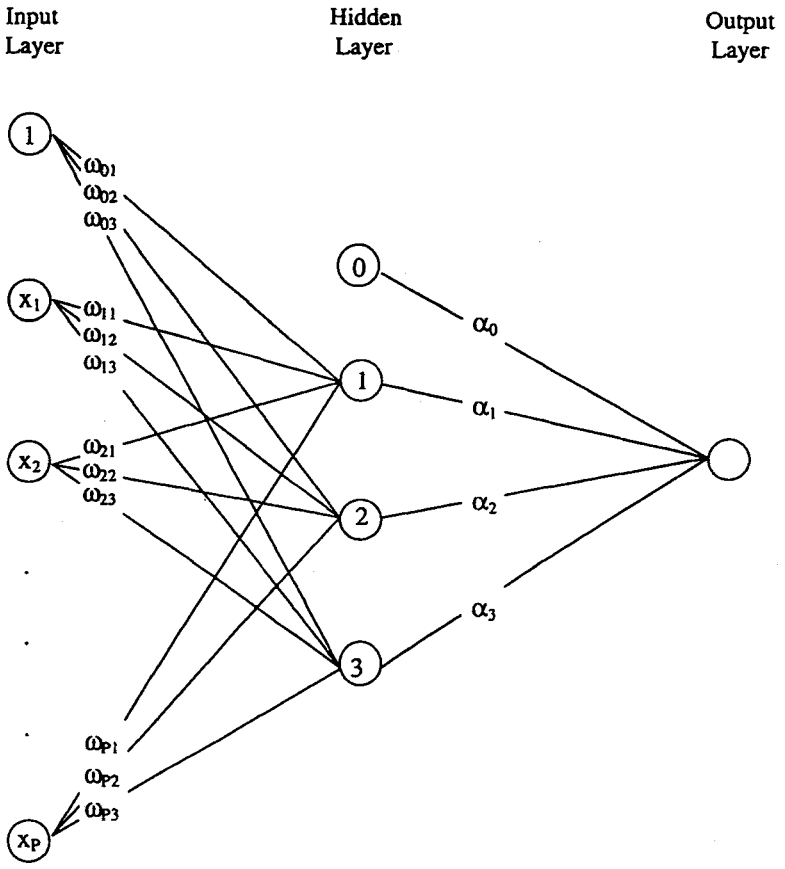

Figure 1. Single Hidden Layer with Three Hidden Nodes Neural Network

den" nodes. An example of such network with 3 nodes in the hidden layer is shown in Figure 1. Each input is connected directly to all but one node in the hidden layer. The output from each hidden node is a non-linear transformation on a weighted sum of the inputs. Every hidden node is connected directly to the output node. No feedback connections are permitted in these "feed-forward" networks. Consider a vector of covariates $x_i = (x_{i0}, x_{il}, \ldots, x_{iP})'$ for the ith case as an input to the network $(i = 1, \ldots, n)$, where x_{i0} is always 1. The pth input node takes value x_{ip} for this input vector $(p = 0, \ldots, P)$. A weight (parameter) w_{ph} is associated with the connection between input p and hidden node h $(h = 1, \ldots, H)$. In general, a neural network has a special hidden node (node 0) connected to the output nodes but not the input nodes. Its role is similar to the constant term in linear regression. However, in the proportional hazards setting it is not needed because its effect is incorporated into the baseline hazard. The vector of weights associated with the inputs to hidden node h will be denoted $w_h = (w_{0h}, w_{1h}, \ldots, w_{Ph})'$. The net input to hidden node h for the ith case is a

linear projection $w_h'x_i$. The output of hidden node h is $f(w_h'x_i)$ where f is a non-linear transformation or a "squashing function". The most commonly used squashing function is the logistic function $f(z) = 1/(1 + \exp(-z))$. When the identity squashing function $f(z) = z$ is used the network output becomes a linear function of the inputs. In this paper we restrict ourselves to the logistic squashing function. We omit the use of a squashing function for the output layer for reasons discussed below. The weights w_{0h} associated with the unity input x_{i0} are called "bias" terms. The output from the network is a weighted sum of the output of the hidden nodes with weights $\alpha_0, \ldots, \alpha_H$. Therefore the functional representation of the output from a single hidden layer neural network with H hidden nodes for a given input vector x_i is

$$g(x_i, \theta) = \alpha_0 + \sum_{h=1}^{H} \alpha_h f(w_h'x_i) = \alpha_0 + \sum_{h=1}^{H} \alpha_h/[1 + \exp(-w_h'x_i)], \qquad (1)$$

where θ denotes the vector of unknown parameters, $(w_{01}, w_{11}, \ldots, w_{P1}, w_{02}, w_{12}, \ldots, w_{P2}, \ldots, w_{0H}, w_{1H}, \ldots, w_{PH}, \alpha_0, \alpha_1, \ldots, \alpha_H)'$. The number of parameters in (1) is $(H + 1) + (P + 1)H$. We define $g(x, \theta) = [g(x_1, \theta), \ldots, g(x_n, \theta)]'$ as the vector of outputs for all cases.

Neural networks are generally "trained" using the back propagation algorithm. Training means determining values for the parameters θ. A training data set therefore must consist of both the inputs and the outputs for a set of cases. The back propagation algorithm iteratively estimates $\hat{\theta}$ via a gradient descent method. The training of the network starts by choosing starting values $\hat{\theta}_0$. An updated estimate is determined as observations are considered one at a time. Multiple passes through the data are usually needed before the estimates converge. For further discussison of back-propagation methods see WASSERMAN (1989).

2.2 *The proportional hazards neural network model*

The proportional hazards model (COX, 1972) is commonly used to model censored data and covariates. The underlying assumption of the model is that the hazard function $h(.)$ depends on time t and a vector of covariates $x_i^* = (x_{i1}, x_{i2}, \ldots, x_{iP})'$, $i = 1, \ldots, n$ through

$$h(t, x_i^*) = h_0(t) \exp\{\beta' x_i^*\}, \qquad (2)$$

where $h_0(t)$ is the baseline hazard independent of the covariates and $\beta = (\beta_1, \ldots, \beta_P)'$. Note that in the proportional hazards model we do not include the constant term as it is incorporated into the baseline hazard. Hence the difference between x_i and x_i^* is that x_i includes $x_{i0} \equiv 1$ and x_i^* does not. The vector of parameters is estimated by maximizing the partial likelihood

$$L_c(\beta) = \prod_{i \in u} \frac{\exp(\beta' x_i^*)}{\sum_{j \in R_{(i)}} \exp(\beta' x_j^*)} \qquad (3)$$

using the Newton-Raphson method. The product in (3) is taken over the uncensored observations and the risk set $R_{(i)}$ is defined by all cases that were alive at each ordered uncensored time $t_{(i)}$. TSIATIS (1981) proved that asymptotically the vector of parameter estimates $\hat{\beta} \mid \beta \to N(\beta, K)$ where K^{-1} is the Fisher information matrix. Inference on the parameters of the proportional hazards model are made replacing the Fisher information matrix with the sample information matrix. For more details see MILLER (1981).

FARAGGI and SIMON (1995a) replaced the linear functional $\beta' x_i^*$ in (2) by the output $g(x_i, \theta)$ of the network. The proportional hazards model becomes

$$h(t, x_i) = h_0(t) \exp [g(x_i, \theta)] \tag{4}$$

and the function to be maximized becomes

$$L_c(\theta) = \prod_{i \in u} \frac{\exp \left\{ \sum_{h=1}^{H} \alpha_h / (1 + \exp (-w_h' x_i)) \right\}}{\sum_{j \in R_{(i)}} \exp \left\{ \sum_{h=1}^{H} \alpha_h / (1 + \exp (w_h' x_j)) \right\}}. \tag{5}$$

Note that in the proportional hazards case, we do not include the constant α_0 and hence the number of parameters in the proportional hazards neural network model is $m = H + H(P+1) = H(P+2)$.

We obtain maximum likelihood estimates (mle's) of the parameters of the proportional hazards neural network using the Newton-Raphson method to maximize the partial likelihood (5) using GAUSS (1993) maximum likelihood software. Both the first and second derivatives are approximated numerically.

2.3 *Bayesian analysis*

Let the vector θ^* denote all the parameters of the proportional hazards model (4). These include the parameters θ and the parameters of the underlying hazard $h_0(t)$, although we are only interested in θ.

EFRON (1977) showed that if $h_0(t)$ is any smooth monotone function, then the mle $\hat{\theta}$ is asymptotically efficient. BEGUN et al. (1983) showed that $\hat{\theta}$ is semi-parametric efficient. OAKES (1977) showed that even in other cases, its asymptotic efficiency is generally quite high (>90%). Consequently, we will base our inference on the posterior distribution of $\theta \mid \hat{\theta}$.

We shall assume that the vector of unknown parameters has prior distribution $\theta \sim N(0, D)$ and shall use the fact that the mle $\hat{\theta} \mid \theta$ is asumptotically $N(\theta, C)$. It follows from LINDLEY and SMITH (1972) therefore that the posterior distribution of $\theta \mid \hat{\theta}$ is approximately $N(Bb, B)$ where $B^{-1} = C^{-1} + D^{-1}$ and $b = C^{-1}\hat{\theta}$. This posterior distribution is used as the basis for our Bayesian analysis. This approach avoids the need to place a prior distribution on the baseline hazard function $h_0(t)$

and it permits us to estimate any function of θ. Our result is a large sample approximation based on the asymptotic normality of the mle and ignoring the variability in the sample information matrix as an estimate of Fisher information.

For simplicity we chose the prior covariance matrix to be $D = \sigma_0^2 I$ where I is an identity matrix. In practice this requires the centering and scaling of the terms of the model (DUMOUCHEL and JONES, 1994). We vary the parameter σ_0^2 of the prior distribution to examine how robust the results are to these changes.

As $\sigma_0^2 \to \infty$, the posterior mean $Bb = (C^{-1} + D^{-1})^{-1} C^{-1}\hat{\theta} \to \hat{\theta}$, i.e. the posterior mean is reduced to the usual maximum likelihood estimators of θ. Because the prior mean is zero, the posterior mean represents a shrinkage of the maximum likelihood estimators towards zero. The degree of shrinkage depends on the prior variance chosen. For fully parametric models in which the parameter vector θ has a $N(0, \lambda^{-1} I)$ prior distribution, the marginal log likelihood has the form

$$\log L(\theta) + \lambda \sum \theta_k^2 , \tag{7}$$

where $L(\theta)$ is the likelihood given (θ). It is this penalized function which has been used by RIPLEY (1994) and others to limit over-fitting of neural networks.

The relationship between a penalized likelihood and Bayesian estimators is well documented in the statistical literature mostly for the normal linear model and ridge regression (DRAPER and SMITH, 1982). One can also verify that for the normal linear case when the observations $Y \sim N(X\gamma, \sigma^2 I)$ so that $\hat{\gamma} \mid \gamma \sim N(\gamma, \sigma^2 (X'X)^{-1})$, assuming prior $\gamma \sim N(0, \sigma_0^2)$ the posterior mean is $[X'X + (\sigma^2/\sigma_0^2) I]^{-1} X'Y$ i.e. posterior mean reduces to the ridge regression estimator. Also in the neural network literature several authors used a Bayesian formulation to obtain shrunken estimators and showed the equivalence with the penalized likelihood (e.g. MACKAY, 1992), however all minimized the squared error function corresponding to a normal likelihood.

Our approach is different, it does not depend on normal likelihood and can be extended to other neural network models where mle's are obtained, for example the logistic likelihood for classification problem (FARAGGI and SIMON, 1995b). Furthermore, using our suggested approximation once the maximum likelihood estimators are obtained together with the sample information matrix we can calculate, rather than re-estimate, the posterior mean for any specified σ_0^2 (or for any matrix D in general). These calculations rather than estimations reduce the computation time dramatically.

3. Simulated example

Survival times were generated from the exponential distribution with hazard $\delta_1 X_1 + \delta_2 X_2 + \delta_3 X_3 + \delta_{12} X_1 X_2 + \delta_{13} X_1 X_3 + \delta_{23} X_2 X_3$ where $(\delta_1, \delta_2, \delta_3, \delta_{12}, \delta_{13}, \delta_{23})' = (0.5, 0.7, 0.87, 1.3, 1.4, 0.72)'$ and the covariates were $X_1 \sim$ Bernulli (0.5), $X_2 = \ln\left(\frac{2}{\tilde{X}_2 + 0.01}\right)$ where $\tilde{X}_2$ is a uniform variable on the interval [0, 1] and

$X_3 = \sqrt{\tilde{X}_3}$ where $\tilde{X}_3$ is the absolute value of a standard normal variable. 20% random censoring was imposed on the generated survival times. Four hundred observations were generated and were randomly split into 2 groups of 200 each. One group was used for the training process and the other for validation.

To evaluate the procedures we used neural network proportional hazards models with varying number of hidden nodes. As inputs to the neural network we used X_1, $\tilde{X}_2$, $\tilde{X}_3$ and two other standard normal random variables X_4 and X_5 to serve as noise variables. This gave six nodes in the input layer. The models that were compared were:

I. Neural network model with 2 hidden nodes (14 parameters).
II. Neural network model with 3 hidden nodes (21 parameters).
III. Neural network model with 4 hidden nodes (28 parameters).
IV. Neural network model with 5 hidden nodes (35 parameters).
V. Neural network model with 6 hidden nodes (42 parameters).

Two methods of parameter estimation were used; maximum likelihood estimators (mle's) $\hat{\theta}$ and the posterior mean of the parameters. Mle's were obtained by training the models using the MAXLIK procedure of GAUSS (1993). Bayesian estimators were computed using the posterior mean $Bb = (C^{-1} + \sigma_0^{-2} I)^{-1} C^{-1} \hat{\theta}$, where C^{-1} is the sample information matrix, varying the values of $\sigma_0^2 = 0.0001$, 0.001, 0.01, 0.1, 0.5, 1 and 2. All parameters were estimated on the training set.

Model performance was evaluated based on the value of the log-partial likelihood and the c index in the validation data-set. The c index can be regarded as the proportion of predictions that are concordant out of all pairs of observations for which ordering of the survival times can be determined (HARRELL et al., 1984). A concordant prediction refers to a pair of observations in which the observation with higher probability of survival is also observed to survive longer.

Table 1

Simulated Example; Log Likelihood and c Index Values for the Neural Network Model with 2 Hidden Nodes Both for the Training and Validation Sets

Model	Log-likelihood		c-Index	
	Training	Validation	Training	Validation
True	−608.2	−598.9	0.776	0.725
Bayes; $\sigma_0^2 = 0.0001$	−681.9	−642.7	0.731	0.659
$\sigma_0^2 = 0.001$	−678.9	−640.9	0.762	0.689
$\sigma_0^2 = 0.01$	−653.2	−628.2	0.789	0.707
$\sigma_0^2 = 0.1$	−601.0	−611.8	0.791	0.705
$\sigma_0^2 = 0.5$	−593.9	−618.9	0.791	0.704
$\sigma_0^2 = 1.0$	−593.5	−620.9	0.791	0.704
$\sigma_0^2 = 2.0$	−593.4	−622.1	0.792	0.704
M.L.E.	−593.4	−623.3	0.792	0.704

Table 2
Simulated Example; Log Likelihood and c Index Values for the Neural Network Model with 3 Hidden Nodes Both for the Training and Validation Sets

Model	Log-likelihood		c-Index	
	Training	Validation	Training	Validation
True	−608.2	−598.9	0.776	0.725
Bayes; $\sigma_0^2 = 0.0001$	−681.9	−642.6	0.712	0.688
$\sigma_0^2 = 0.001$	−677.6	−639.2	0.749	0.687
$\sigma_0^2 = 0.01$	−638.3	−616.4	0.737	0.651
$\sigma_0^2 = 0.1$	−586.0	−626.9	0.732	0.613
$\sigma_0^2 = 0.5$	−580.6	−654.2	0.731	0.613
$\sigma_0^2 = 1.0$	−580.4	−659.7	0.731	0.612
$\sigma_0^2 = 2.0$	−580.3	−662.8	0.731	0.612
M.L.E.	−580.3	−666.0	0.731	0.612

Results are summarized in Tables 1–5 where each table presents the results for a different model. Each table contains the 2 measures for both training and validation parts. We present the results from eight different estimators namely the mle and the posterior mean using seven different values of σ_0^2. In order to provide a baseline for comparison, we calculated the value of the partial log likelihood and the c index for the "true" model. For the "true" model, the partial log likelihood and the c index were computed using the constants $\delta = (\delta_1, \delta_2, \delta_3, \delta_{12}, \delta_{13}, \delta_{23})'$ and the matrix X that were used to generate the simulated example.

Analyzing the results in the training set we observe that the log-likelihood of the true model is −608.2. Due to over-fitting in the training data, the mle's provide much higher values of the log-likelihoods, increasing from −593.4 for the 2 hidden nodes model (Table 1), to −552.7 for the 6 hidden nodes model (Table 5).

Table 3
Simulated Example; Log Likelihood and c Index Values for the Neural Network Model with 4 Hidden Nodes Both for the Training and Validation Sets

Model	Log-likelihood		c-Index	
	Training	Validation	Training	Validation
True	−608.2	−598.9	0.776	0.725
Bayes; $\sigma_0^2 = 0.0001$	−680.6	−641.9	0.784	0.708
$\sigma_0^2 = 0.001$	−668.2	−634.2	0.794	0.706
$\sigma_0^2 = 0.01$	−616.4	−619.9	0.743	0.633
$\sigma_0^2 = 0.1$	−590.6	−644.9	0.722	0.616
$\sigma_0^2 = 0.5$	−577.7	−632.5	0.724	0.625
$\sigma_0^2 = 1.0$	−576.5	−629.6	0.724	0.623
$\sigma_0^2 = 2.0$	−576.1	−628.2	0.724	0.623
M.L.E.	−576.0	−626.7	0.724	0.623

Table 4

Simulated Example; Log Likelihood and c Index Values for the Neural Network Model with 5 Hidden Nodes Both for the Training and Validation Sets

Model	Log-likelihood		c-Index	
	Training	Validation	Training	Validation
True	−608.2	−598.9	0.776	0.725
Bayes; $\sigma_0^2 = 0.0001$	−682.0	−642.7	0.622	0.657
$\sigma_0^2 = 0.001$	−677.0	−638.7	0.780	0.703
$\sigma_0^2 = 0.01$	−641.3	−616.2	0.786	0.664
$\sigma_0^2 = 0.1$	−577.3	−629.5	0.754	0.592
$\sigma_0^2 = 0.5$	−568.2	−667.2	0.750	0.584
$\sigma_0^2 = 1.0$	−567.7	−675.4	0.749	0.584
$\sigma_0^2 = 2.0$	−567.5	−679.9	0.749	0.583
M.L.E.	−567.5	−684.8	0.749	0.583

This over-fitting is shown also in the validation set where the true model achieves a log likelihood value of −598.9 while the neutral network models values range from −623.3 for the model with 2 hidden nodes to −3314 for the model with 6 hidden nodes. Note that for the validation set indication of over-fit is much *lower* values of the log-likelihood compared with the true model. This is the case because the parameters are estimated on the training set and when they are applied to the validation set they generalize poorly and produce a log-likelihood value that is much lower than the true value.

Furthermore, for the validation set, the results of the posterior mean with $\sigma_0^2 = 0.0001$ for the different neural network models are very close. The log-likelihood obtained is about −642 for all models. This implies that this prior variance

Table 5

Simulated Example; Log Likelihood and c Index Values for the Neural Network Model with 6 Hidden Nodes Both for the Training and Validation Sets

Model	Log-likelihood		c-Index	
	Training	Validation	Training	Validation
True	−608.2	−598.9	0.776	0.725
Bayes; $\sigma_0^2 = 0.0001$	−681.9	−642.7	0.629	0.597
$\sigma_0^2 = 0.001$	−676.1	−639.6	0.695	0.631
$\sigma_0^2 = 0.01$	−627.6	−663.0	0.794	0.665
$\sigma_0^2 = 0.1$	−561.5	−2221	0.822	0.657
$\sigma_0^2 = 0.5$	−553.4	−3017	0.826	0.659
$\sigma_0^2 = 1.0$	−552.9	−3158	0.826	0.659
$\sigma_0^2 = 2.0$	−552.8	−3234	0.826	0.659
M.L.E.	−552.7	−3314	0.826	0.659

is too small and all the parameters are shrunken toward zero so that no difference between the models is observed and all the models are similar to the model with no parameters. With prior value $\sigma_0^2 = 2$ the values obtained for the posterior mean are very close to the values obtained from the mle's, which implies that this prior value is too large. Between these two extremes one can see a similar trend over the different models. Observing the validation set we find that the log likelihood is improving from about -642 to some middle prior value from which the log-likelihoods decrease back towards the log-likelihood of the mle. A similar trend, though much weaker, is obtained from the *c*-statistic. At some prior value between $\sigma_0^2 = 0.0001$ and $\sigma_0^2 = 2$, the measures of fit for Bayesian estimation reach their maximum values. Prior values which produce best log-likelihoods were: $\sigma_0^2 = 0.1$ for 2 hidden nodes model. $\sigma_0^2 = 0.01$ for 3, 4, 5 hidden nodes models. $\sigma_0^2 = 0.001$ for 6 hidden nodes model.

Maximum value of log-likelihood in the validation set (-611.8) was obtained from the 2 hidden nodes model using Bayesian estimation with prior value of $\sigma_0^2 = 0.1$ indicating that 2 hidden nodes are enough. The six hidden nodes model, although much more complex (42 parameters), does not improve the fit of the model beyond that of the two hidden nodes model (only 14 parameters). It is also seen that as we increase the complexity of the model (by increasing the number of hidden nodes to the neural network model), one needs to set a smaller prior value to the covariance matrix for the Bayesian estimation. Smaller prior values mean greater restriction on estimation, which is needed to reduce over-fitting.

4. An Example

To illustrate the Bayesian Neural Network model we have used a subset of data collected for the Intenational Germ Cell Cancer Collaborative Group relating to the development of new international staging system for metastatic germ cell cancer. Ten nations contributed data for the analysis of prognostic factors in male patients treated with platinum based chemotherapy between 1975 and 1990. Median follow up was 5 years. MEAD (1995) analyzed the data and reported 5 major prognostic factors for time to tumor progression. (a). Mediastinal primary site. (b). nonpulmonary visceral metastases. (c). AFP-alpha-fetprotein. (d). hCG – human chorionic gonadotropin. (e). LDH – Lactic dehydrogenase. The first two covariates were binary while the other were continuous covariables. All the continuous variables were transformed to the base 10 logarithm and then centered and scaled. We have randomly split the data into 3 subsets, each sub-set consists of 350 observations. The distributions of the covariates were similar in these three subsets. Parameter estimation of the neural networks was done on the first (training) set by reaching maximum likelihood. Once the mle's for the neural network parameters were obtained the posterior means *Bb* were calculated for different σ_0^2. With these posterior means we evaluated the log-likelihood in a second set to determine the optimal value of σ_0^2. The third set was used for validation.

Table 6
An Example; Log-Likelihood values for the Bayesian Neural Network Models and M.L.E.'s

Model	2 Hidden Nodes			4 Hidden Nodes			6 Hidden Nodes		
σ_0^2	Training	Second	Validation	Training	Second	Validation	Training	Second	Validation
0.0001	−1026.0	−910.7	−1073.3	−1021.0	−905.6	−1066.1	−1025.5	− 910.8	−1173.4
0.001	−1019.6	−904.9	−1066.5	−1003.7	−892.2	−1043.9	−1014.0	− 905.9	−1068.2
0.01	− 977.1	−867.9	−1021.8	− 965.7	−873.7	−1028.9	− 945.6	− 883.7	−1037.9
0.1	− 960.2	−854.8	−1014.5	− 944.1	−882.7	−1054.5	− 921.6	− 973.1	−1095.6
0.5	− 959.9	−855.8	−1016.5	− 943.0	−889.6	−1067.6	− 914.7	−1015.8	−1120.5
1.0	− 959.9	−856.0	−1016.9	− 943.0	−890.8	−1069.7	− 914.5	−1022.5	−1124.4
2.0	− 959.9	−856.1	−1017.0	− 943.0	−891.4	−1070.8	− 914.4	−1026.0	−1126.5
M.L.E.	− 959.9	−856.2	−1017.2	− 943.0	−892.0	−1071.9	− 914.4	−1029.6	−1128.6

Table 6 provides the results for the different Bayesian neural network models as well as the mle's. For brevity we provide only the results of the neural network models with 2, 4, and 6 hidden nodes since the other models showed very similar results. Table 6 indicates that for any neural network model, while obviously for the training set the log-likelihood value reaches its maximum with the mle's, the results for the second and the validation sets were poor for the maximum likelihood estimates, an indication of over-fit. As for the Bayesian neural network models, for any given number of hidden nodes, σ_0^2 can be found that maximizes the log-likelihood for the second set. For example, for the neural network model with 6 hidden nodes, the posterior mean that was obtained for $\sigma_0^2 = 0.01$ maximized the log-likelihood for the second set. This model also maximized the log-likelihood for the validation set.

5. Discussion

We have compared the performance of Bayesian estimation to maximum likelihood estimation in the neural network proportional hazards model. Extremely large estimators usually imply that estimation has reached a stage of over-fitting. Hence, one way to avoid over-fitting is by controlling and limiting the magnitude of the estimators.

The prior variance σ_0^2 controls the amount of shrinkage applied to the mle's depending on the complexity of the data. In the simulated example presented in this paper the maximum value of the log likelihood in the validation set was obtained from the 2 hidden nodes model with $\sigma_0^2 = 0.1$. Another indication that no more than two hidden nodes were needed is observing that the best values of σ_0^2 decreased as we added more hidden nodes i.e. greater shrinkage as more parameters were added. Also in our germ cell cancer example the model with 2 hidden nodes and $\sigma_0^2 = 0.1$ is the best model. In other situations, however, more hidden nodes might be needed. Since obtaining results for a range of prior variances is computationally fast, we recommend that such a search be done and both the optimum number of hidden nodes and the value of σ_0^2 be determined from the second set. Finally, if a sufficient number of observations are available the model should be checked on a third independent data set. If not enough observations are available resampling methods like cross-validation (EFRON, 1982) can be applied to verify the generalization of the results.

Bayesian formulations of neural networks have been proposed previously by several authors. KONONENKO (1989) and YEUNG (1993) used Bayesian neural networks for classification problems using back-propagation (RUMELHART, HINTON, and WILLIAMS, 1986) to minimize the squared error function. MACKAY (1995) suggested the Bayesian framework to interpret neural network models and parameters and a method of comparing models. MacKay also used Bayesian models with back-propagation to minimize a penalized sum of squares function.

For censored survival data LIESTOL, ANDERSEN, and ANDERSEN (1994) suggested using the back-propagation algorithm on modification of the proportional hazard models. They considered two modifications for grouped survival data with a piecewise constant baseline hazard. They found that to improve prediction ability they had to add a penalty to the squared error function that the back-propagation algorithm minimized. Our approach differs in that we use the proportional hazards model and the partial likelihood (COX, 1972) to obtain the mle's. Once they are found we calculate the posterior mean for any pre-specified prior variance and thereby control the amount of shrinkage.

It is important to emphasize that our goal here is to use a computationally efficient Bayesian approach to improve predictiveness in the proportional hazards neural network model proposed by FARAGGI and SIMON (1995a). It was not our intention to compare the neural network approach to the linear proportional hazards model.

Acknowledgement

The authors acknowledge the referee for making useful comments that improved the manuscript.

References

BEGUN, J. M., HALL, W. J., HUANG, W. M., and WELLNER, J. A., 1983: Information and Asymptotic Efficiency in Parametric-Nonparametric Models. *The Annals of Statistics* **11**, 432–452.

BUCKELY, J. and JAMES, I., 1979: Linear regression with censored data. *Biometrika* **66**, 429–436.

BUNTINE, W. L. and WEIGEND, A. S., 1991: Bayesian Back-Propagation. *Complex Systems* **5**, 603–643.

COX, D. R., 1972: Regression Models and Life Tables. (with discussion). *Journal of the Royal Statistical Society* B **34**, 187–220.

DAPONTE, J. S. and SHERMAN, P., 1991: Classification of ultrasonic image texture by statistical discriminate analysis of Neural Network. *Comput. Med. Image. Graph.* **15**, 3–9.

DRAPER, N. R. and SMITH, H., 1982: *Applied Regression Analysis*, 2nd edition. Wiley, New York.

DUMOUCHEL, W. and JONES, B., 1994: A Simple Bayesian Modification of D-Optimal Designs to Reduce Dependence on an Assumed Model. *Technometrics* **36**, 37–47.

EFRON, B., 1977: The efficiency of Cox's likelihood function for censored data. *Journal of the American Statistical Association* **72**, 557–565.

EFRON, B., 1982: *The Jackknife the Bootstrap and Other Resampling Plans*. SIAM, Philadelphia.

FARAGGI, D. and SIMON, R., 1995a: A Neural Network Model for Survival Data. *Statistics in Medicine* **14**, 73–82.

FARAGGI, D. and SIMON, R., 1995b: The Maximum Likelihood Neural Network as a Statistical Classification Problem. *Journal of Statistical Planning and Inference* **46**, 93–104.

GALLANT, A. R. and WHITE, H., 1991: On Learning the Derivatives of an Unknown Mapping With Multilayer Feedforward Networks. *Neural Networks* **5**, 129–138.

Aptech Systems, Inc., 1993: *GAUSS Applications: Maximum Likelihood*. Aptech Systems, Maple Valley, WA.

GEMAN, S., BIENSTOCK, E., and DOURSAT, R., 1992: Neural Networks and the Bias/Variance Dilemma. *Neural Computation* **4**, 1–58.

HARRELL, F. E., LEE, K. L., CALIFF, R. M., PRYOR, D. B., and ROSATI, R. A., 1984: Regression modelling strategies for improved prognostic prediction. *Statistics in Medicine* **3**, 143–152.

HOLLEY, L. H. and KARPLUS, M., 1989: Protein Structure Prediction with a Neural Network. *Proc. Natl. Acad. Sci., U.S.A.* **86**, 152–156.

KONONENKO, I., 1989: Bayesian Neural Network. *Biological Cybernetics* **61**, 361–370.

LIESTOL, K., ANDERSEN, P. K., and ANDERSEN, U., 1994: Survival Analysis and Neural Nets. *Statistics in Medicine* **13**, 1189–1200.

LINDLEY, D. V. and SMITH, F. M., 1972: Bayes Estimates for the Linear Model. (with discussion). *Journal of the Royal Statistical Society* B **34**, 1–41.

MACKAY, D. J. C., 1992: A Practical Bayesian Framework for Backpropagation Networks. *Neural Computation* **4**, 448–472.

MACKAY, D. J. C., 1995: Bayesian Neural Networks and Density Networks. *Nuclear Instruments & Methods in Physics Research A* **354**, 73–80.

MANN, N. H. I. and BROWN, M. D., 1991: Artificial Intelligence in the Diagnosis of Low Back Pain. *Orthop. Clin. North Am.* **22**, 303–314.

MEAD, G. M. on behalf of the IGCCCG, 1995: *International Consensus Prognostic Classification for Metastatic Germ Cell Tumors Treated with Platinum Based Chemotherapy.* A Final Report of the International Germ Cell Cancer Collaborative Group (IDCCCG). ASCO.

MILLER, R. G., 1981: *Survival Analysis.* Wiley, New York.

OAKES, D., 1977: The asymptotic Information in Censored Survival Data. *Biometrika* **64**, 441–448.

PLAUT, D. C., NOWLAN, S. J., and HINTON, G. E., 1986: Experiments on Learning by Back-Propagation. Technical Report CMU-CS-86-126, Carnegie Mellon University, Pittsburgh, PA 15213.

PRENTICE, R. L. and KALBFLEISCH, J. D., 1979: Hazard rate models with covariates. *Biometrics* **35**, 25–39.

RIPLEY, B. D., 1994: Statistical Aspects of Neural Networks. In: O. Barndorff-Nielsen, J. L. Jensen, and W. S. Kendall (Eds.): *Networks and Chaos – Statistical and Probabilistic Aspects.* Chapman and Hall, London.

RUMELHART, D. E., HINTON, G. E., and WILLIAMS, R. J., 1986: Learning Internal Representations by error Propagation. In: D. E. Rumelhart: *Parallel Distributed processing* – Vol. 1. MIT Press, Cambridge, MA, 318–362.

SEJNOWSKI, T. J. and ROSENBERG, L. R., 1987: Parallel networks that learn to pronounce english text. *Complex Systems* **1**, 145–168.

TSIATIS, A., 1981: A Large Sample Study of Cox's Regression Model. *Annals Of Statistics* **9**, 93–108.

WASSERMAN, P. D., 1989: *Neural Computing Theory and Practice.* Van Nostrand Reinhold, New York.

WEINSTEIN, J. N., KOHN, K. W., GREVER, M. R., VISWANADHAN, V. N., RUBINSTEIN, L. V., MONKS, A. P., SCUDIERO, D. A., WELCH, L., KOUTSOUKOS, A. D., CHIAUSA, A. J., and PAULL, K. D., 1992: Neural computing in cancer drug development predicting mechanism of action. *Science* **258**, 447–451.

YEUNG, D.-Y., 1993: Constructive Neural Networks as Estimator of Bayesian Discriminant Functions. *Pattern Recognition* **26**, 189–204.

DAVID FARAGGI
Department of Statistics
Faculty of Social Sciences and Mathematics
University of Haifa
Mount Carmel
Haifa 31905
Israel

Received, October 1996
Revised, June 1997
Accepted, June 1997