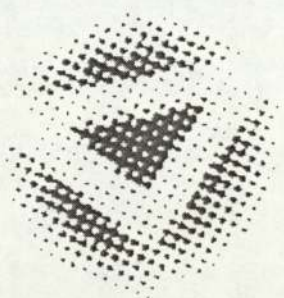
Survival Data Analysis Using Neural Networks

Elena Ellioti

MSc by Research in Pattern Analysis and Neural Networks



The University of Aston in Birmingham

September 1997

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without proper acknowledgement.

THE UNIVERSITY OF ASTON IN BIRMINGHAM

Survival Data Analysis Using Neural Networks

ELENA ELLIOTI

MSc by Research in Pattern Analysis and Neural Networks, 1997

Thesis Summary

This thesis introduces a new approach in survival data analysis i.e., Cox regression using neural networks. It is implemented on a data set of 575 patients which was provided by the CRC trials at Birmingham University.

In the first part of the project, the standard Cox regression method was implemented. The objective of Cox regression is to model the probability functions of the patients based on a fundamental hypothesis which is described in the thesis. First the method was implemented on synthetic data in order to check its performance. Then it was implemented on the real data, and the results were compared with the ones found by the statisticians at the CRC trials.

In the second part the new approach was introduced: Cox regression using neural networks. It was again applied first to synthetic data, and then to the real data. The results obtained were compared with the ones found by the implementation of standard Cox method.

In the third and last part, the cumulative baseline hazard function was estimated. After applying the method to synthetic data, as before, it was implemented on the real data, and the cumulative hazard function and survival probability were also estimated. The implementation was done using both approaches and the results were compared. The conlcusions derived from the results obtained are discussed at the end of the thesis.

Keywords: Cox regression, Cox regression using neural networks, Prognostic factors, Cumulative hazard, Cancer

Acknowledgements

I would like to thank my supervisor Christopher Bishop for his constant help, guidance and patience. I would also like to thank Christopher Williams for his help and patience, as towards the end of the fourth term he became my supervisor.

Many thanks to Jannet Dunn and Debbie Mofatt, of the CRC trials in Birmingham University, for their great help in answering my questions concerning Cox method and the data set, and for their kindness. I would also like to thank the MSc pann students for their good humour and support, and in general all the Post Docs and PhD students.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The introduction includes background information about the problem, a description of the data set used, and the way that the data was pre-processed for the purpose of this project.

1.1 The problem

1.1.1 Overview of the problem

The project deals with survival data analysis. In survival data analysis, interest centres on a group or groups of individuals (or objects) for each of whom (or which) there is a specific time event, normally called failure, which occurs after a period of time called the failure time. Failure can occur at most once for any individual of the group (Cox & Oakes [1]).

Examples of failure times include the lifetimes of machine components in industrial reliability, the durations of strikes or periods of unemployment in economics, the times taken by subjects to complete specified tasks in psychological experimentation and the survival times of patients in clinical trials, which is the topic under consideration in this project.

We are concerned with the problem that clinicians have to face when treating patients with cerebral metastases. These patients have had a primary tumour elsewhere in the body and then cancer metastases in the brain.

The aim of the project is to help statisticians to predict the survival times of

the patients, given their characteristics. This will be achieved by introducing neural networks in addition to the standard methods used up to now. All the results are scientifically confirmed and tested.

1.1.2 Background of the problem

There has been a lot of study of patients that develop cerebral metastases by clinicians.

The location of metastases within the brain has been analysed. The parietal lobe is the most common site that the tumour appears. Multiple sites follow: frontal lobe, temporal lobe, occipital lobe and cerebellum, in that order. Other locations with occasional metastases are the mid-brain, hypothalamus and ventricles. There are also some cases where the location of the brain tumour cannot be specified (J. West et al [11]).

Lung cancer is the most frequent primary site, comprising 50% or more in the case of most series, breast is the second most common primary side with a reported frequency that varies greatly (10 - 45%), depending on the population under study (Lawrence et. al [12]).

The diagnosis and management of patients with cerebral metastases and the techniques for irradiation have slowly evolved over the past four decades. Major factors in this evolution include improved brain imaging through the development of CT scanning and MRI, and improved knowledge of optimal dose fraction schedules determined largely by randomised trials conducted by the Radiation Therapy Oncology Group (RTOG) (Lawrence et al [12]).

A study that took place in Texas showed that cancer metastases in the brain are usually multi-focal, even though diagnostic tests may not demonstrate multiplicity. It has been stated that approximately 80% of cases actually involve multiple metastases (Borgelt et. all [10]). Of course, the form of the brain tumour is not the only factor that influences the survival time of the patient. There are several other factors that have a big contribution. As the statisticians at the CRC trials Unit in Birmingham University have stated, the actual form of the tumour is not a very important factor. A patient who has multiple tumours spread around in the brain may be better off than a patient who has one solitary tumour. What is important in this case is the position

of the solitary or the multiple tumours and how that position affects the condition of the patient.

Possible factors influencing the survival time of the patients are: their age, their sex, the treatment they are currently receiving, the extent of their condition, and the form and the position of the primary tumour. The last factor is quite important because there are various kinds of primary tumours that can be treated depending on their position. For example a tumour appearing on the breast is easier to handle than one that appears on the liver.

As clinicians state, determining the best treatment for the patients with cerebral metastases is very difficult to achieve. Part of the difficulty arises because metastatic cancer of the brain is not a single entity but rather a spectrum of diseases with differing natural histories relating to site and aggressiveness of the primary tumour. Also, as stated before, the clinical course of brain metastases depends greatly on the general condition of the patient, the results of the treatment of the primary tumour, complications and the presence or absence of other distant metastases.

The major concern of clinicians is to find an effective treatment for each patient. Usually the patients suffering from cerebral metastases now undergo radiotherapy. The importance of radiotherapy in the treatment of metastatic brain cancer was first reported by Chao et al in 1954 at the Thomas Jefferson University in Philadelphia, and later by Chu and Hilaris in 1961 at the same university (Borgelt et al [10]). They used conventional fractionation regimes to a total whole-brain dose of 3000 to 4000 rad in 3 to 4 weeks: they noted good palliation of symptoms in 60 to 80% of their patients, and a mean survival of 6.6 to 8.2 months in those who responded. Since then, others have confirmed the value of palliative whole-brain irradiation in patients with metastases. Hindo et al at the Hahnemann Medical College in Philadelphia and Shehata et al at the Columbia Presbyterian Medical Centre in New York showed that higher increment doses given in fewer fractions may be as effective as standard fractionation schemes.

A number of factors may be used to predict the degree of response to the irradiation, or its duration, or both: their roles remain unclear. These factors include: primary site and extent of metastases, status of the primary site, neurologic status, general functional status and whether corticosteroids are used.

Radiotherapy is the method of treatment used in the trial that we are concerned with in this project. It is described in detail in the next section.

1.2 Description of the data set

A data set of 553 patients was provided by the CRC trials unit at Birmingham University. All the patients included in the set have developed cerebral metastases. From those 553 patients we excluded the ineligible patients, i.e. those who are still alive, as well as those for whom the data is not complete. The patients who are still alive are only seven and that is why they were excluded from the data set. So, finally, we have a data set of 475 patients. We know eight characteristics of these patients, which are presented in table 1.1. These eight characteristics are called prognostic factors in the survival data analysis literature.

VARIABLES	VALUES
Treatment	2 fractions 10 fractions
Sex	Male Female
Age	number of years
Primary Tumour	bronchus - small cell bronchus - other breast other not known
Extent	Solitary Multiple
Dexamethasone	number of milligrams
WHO index	WHO = 0 WHO = 1 WHO = 2 WHO = 3 WHO = 4
Survival Times	number of days

Table 1.1: Prognostic Factors

An explanation of the prognostic factors is given below:

- **Treatment**: All the patients in this trial had undergone radiotherapy. As mentioned earlier, radiotherapy is the most effective treatment used up to now for treating patients with cerebral metastases. The patients of this trial were split into two groups. The split was well balanced according to the patients characteristics. The first group, which consisted of 270 patients was, receiving 12 Gy (Grays) of radiation in two fractions on consecutive days. The other group with 263 patients was taking 30 Gy in ten daily doses. Treatment was given using parallel opposed fields to the whole brain with the total dose calculated at the mid-plane.

  The object of the study carried out was to determine whether a short fractionation schedule was as effective as a more conventional, longer regimen in the management of patients with symptomatic cerebral metastases.

  Clinical assessment of neurological symptoms and performance status was carried out 4, 8 and 12 weeks after commencing radiotherapy and at 3-monthly intervals thereafter until death. The response was assessed at 4 weeks, 12 weeks and 3-months thereafter. The response was defined as an improvement in at least one of the neurological symptoms without deterioration of any of the other neurological symptoms or signs, or the development of new a neurological deficit. This response had to last for a minimum of 4 weeks. The control of previously uncontrolled fits was also accepted as evidence of an objective response.

- **Sex**: A binary variable stating whether the patient is a male or a female.

- **Age**: This variable indicates the age of the patient when entering the trial and is measured in years. It is calculated as: Date of Entry (DOE) - Date of Birth (DOB).

- **Primary Tumour**: This is the location in the body where the first tumour appeared. There are five cases as shown in table 1.1. It is important to know the position of the primary tumour because it greatly affects the condition of the patient after the cerebral metastases.

- **Extent**: This variable denotes the condition of the brain tumour. It is binary and defines whether the metastases are "solitary" or "multiple". Solitary implies that there is only one tumour in the brain, and multiple means that there are several tumours spread around in the brain. The condition of the patient cannot be assesed immediately from this variable because, as stated earlier, one patient that has multiple metastatic tumours in the brain may be better off than another who has one tumour.

- **Dexamethasone**: This variable indicates the number of milligrammes of dexamethasone taken by each patient. It ranges from 1-16 mg and, if a patient requires increasing amounts of dexamethasone, this implies that his/her condition is getting worse.

- **WHO index**: Stands for World Health Organisation index. It ranges from 0 to 4.

  0. Able to carry out any normal activity without restriction.

  1. Restricted in physically strenuous activity, but ambulatory and able to carry out light work.

  2. Ambulatory and capable of self-care, but unable to carry out any work; up and about more than 50% of working hours.

  3. Capable only of limited self-care; confined to bed or chair more than 50% of working hours.

  4. Completely disabled; cannot carry out any self-care; totally confined to bed or chair.

  In this trial, there are no patients with WHO index 4. It ranges only from 0 to 3.

- **Survival Time**: This variable is the most important one in the trial. It states the survival time of each patient. It is measured in days and is calculated as: Date of Death (DDeath) - DOE.

There were several entry criteria for a patient to be accepted in this trial (The Royal College of Radiologists [6]). They included the confirmation of a primary site or, where no obvious primary was apparent, historical confirmation of cerebral metastases;

the presence of symptoms directly referable to cerebral involvement; a stable dose of dexamethasone over the week prior to randomisation; and the provision of informed consent. All patients had to be over 16 years of age, with WHO performance status 0-3 and a neurological status smaller than 4 according to a modified Medical Research Council scale. Patients who had received cytotoxic chemotherapy in the previous 4 weeks were excluded.

1.3 Manipulation of the data set

1.3.1 The log-rank test

In order to be able to perform the necessary tests, the statisticians at the CRC trials units have converted the values of the prognostic factors to binary values. They were interested in which of the patients were considered to be in a good group and which in a bad group, according to their condition. In this way, the treatment for each group would be determined. Therefore, a binary data set would be helpful in detecting the good and the bad values of the prognostic factors, and subsequently to separate the patients into good and bad groups.

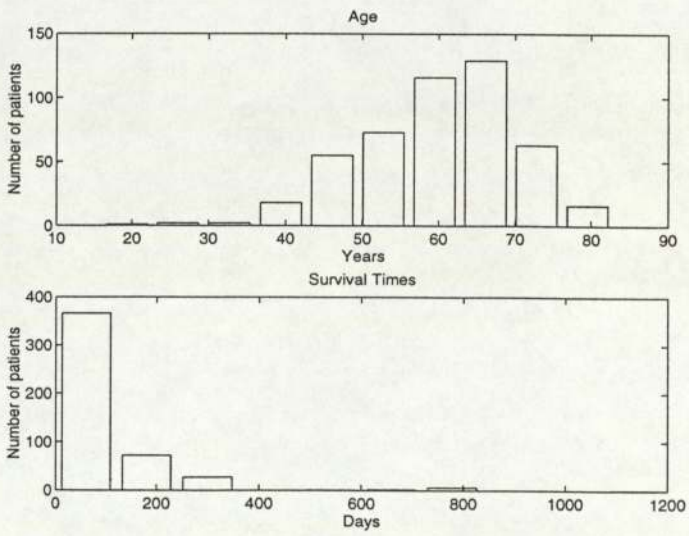


Figure 1.1: Histograms for age and survival times of the patients

For continuous variables like age and survival times, the median was taken as the boundary because in both cases the distribution is skewed, as shown in the histograms

of figure 1.1. As seen from the histograms, there are many patients who are in older ages 55-68 years old, and also many patients who have lower survival times.

They carried out a log-rank test for the variable that include categories, as discussed by Elisa T. Lee in her book (Lee [4]). They found the expected and observed values for each category. The category that had the lowest value was considered good and so it was given the smallest transformed value. For the purposes of this research, it was assigned the value 0.

A brief explanation of the log-rank test is as follows: As stated earlier, this test was carried out only for categorical variables that have more than two categories. The assumption made was that all categories come from the same distribution, so that at each failure time the number of deaths in a given category should be proportional to the size of that category.

The log-rank statistic can be shown to be equal to the sum of the observed failures minus the conditional expected failures computed at each failure time, or simply the difference between the observed and expected failures in one of the groups (category). Let O_i be the observed numbers and E_i be the expected numbers of death in i categories for a specific variable. Since this statistical test is similar to the chi-square test then the value is given by:

$$X_i^2 = \sum_i \frac{(O_i - E_i)^2}{E_i} \tag{1.1}$$

The category with the smallest X^2 value is considered to be the best and, in this case, is given the smallest transformed value, i.e., zero.

To compute E_i we arrange all the observations in ascending order and compute the number of expected deaths at each time and sum them. The number of expected deaths is obtained by multiplying the observed deaths at the time by the proportion of patients exposed to risk in the treatment group. Let d_t be the number of deaths at time t and n_{it} be the number of patients still exposed to risk of dying at time up to t for each category. Then, the expected number of deaths for each category is given by

$$e_t = d_t . \frac{n_{it}}{\sum_i n_{it}} \tag{1.2}$$

Then, the total number of expected deaths in the category is:

Factor	Total	Deaths	Obs/Exp
Treatment			
2 fractions	270	269	1.09
10 fractions	263	257	0.92
Sex			
Male	269	267	1.11
Female	264	259	0.91
Age in years			
<= 60	266	262	0.90
> 60	267	264	1.13
Primary Tumour			
Bronchus small cell	103	103	1.32
Bromchus other	207	204	1.04
Breast	101	98	0.79
Other + Not known	122	121	1.00
Extent			
Solitary	212	208	0.88
Multiple	297	295	1.11
Dexamethasone			
<= 8 mg daily	183	179	0.84
> 8 mg daily	325	323	1.12
WHO index			
0	46	45	0.88
1	149	145	0.80
2	156	155	0.99
3	151	150	1.40

Table 1.2: Obs/Exp values found at the CRC trials unit

$$E_i = \sum_t e_t \tag{1.3}$$

Usually this method, the expectation table is constructed and values of X^2 values are calculated.

The actual observed and expected values are listed in table 1.2. As one can see from this table, the category with a lower obs/exp value also has a lower number of deaths. That is why this category is considered to be the best.Table 1.2 was derived by the statisticians in the CRC trials and they have used 533 eligible patients, that is, they included those patients that have missing data. In our case we do not use the patients

with missing data so that we have, as stated before, 475 patients to consider. Even then the categories determined to be good are found to be the same for the patients under consideration.

1.3.2 The pre-processed data set

The new values that we have set for each prognostic factor and category appear in table 1.3. As mentioned before, the best category takes the value 0 and the rest 1. Appendix A includes the original survival curves of the data by taking into account one prognostic factor at a time. Therefore, in order to verify the results of the log-rank test presented in this section, it is better to inspect the curves in appendix A. This binary data is only used, however, in the calculations of chapter 3, so that the results can be compared with the ones found by the statisticians at the CRC trials unit. To have more accurate and reliable results in the implementation of the new method we use the data set that includes the real values of the prognostic factors.

VARIABLES	VALUES	NEW VALUE
Treatment	2 fractions	1
	10 fractions	0
Sex	Male	0
	Female	1
Age	Below median	0
	Aove median	1
Primary Tumour	breast	0
	bronchus - small cell	1
	bronchus - other	1
	other	1
	not known	1
Extent	Solitary	0
	Multiple	1
Dexamethasone	Below 8mg	0
	Above 8mg	1
WHO index	WHO = 0	0
	WHO = 1	0
	WHO = 2	0
	WHO = 3	1

Table 1.3: New values of the variables

Chapter 2

Cox Regression

In this chapter the Cox regression method is described. This is the method that is mostly used by statisticians for survival data analysis.

2.1 Description of the method

2.1.1 Overview of standard Cox regression method

The basic problem that statisticians are concerned with is the prediction of the survival times of the patients, and the goal is to model the distribution of the survival times within a given population, conditional on the prognostic factors. The statistical approaches used for this problem are generally based on Cox regression. The statisticians at the CRC trials units in Birmingham University that provided us with the data set also use Cox regression.

Cox regression is a semi-parametric technique which combines a log-linear model with the key assumption of proportional hazards (Bishop et al [7]). Let us assume a very basic example to start with. We have a model whose included particles have the following characteristics:

- they are identical, which means they have exactly the same behaviour; and
- they remain unchanged over time until they decay.

Therefore, the probability that a given particle of this model will have a survival time T which is greater than t is given by the well known exponential function:

$$P(T > t) = \exp(-ht) \tag{2.1}$$

where h is called the hazard rate and is a characteristic of the particular type of the particle.

For the case considered in this project, things are different. The particles in the model concerned are the patients suffering from cerebral metastases. So neither of the characteristics described above are valid. Each patient's survival time depends on his/her characteristics and one patient does not have the same values of prognostic factors as the other. So the hazard rate will now depend on both the prognostic factors, a vector x, and a function of time t and it will be denoted as $h(x,t)$ which is the hazard function. It depends on t as well, because h is not constant over time (as it was at the simple example assumed in the beginning of this chapter). The probability of dying from cancer is not constant over time but, in fact, increases with time.

The formal definition of the hazard rate $h(x,t)$ is that it represents the probability per unit time of a patient surviving until time t and then dying before $t + \delta t$. The survival time T of the patient satisfies $t < T < t + \delta t$, and the hazard function is given by:

$$h(x,t) = \lim_{\delta t->0} \frac{P(T < t + \delta t | T > t)}{\delta t}$$

$$= \lim_{\delta t->0} \frac{P(T < t + \delta t, T > t)}{P(T > t)\delta t} \tag{2.2}$$

$$= \lim_{\delta t->0} \frac{P(t < T < t + \delta t)}{P(T > t)\delta t} \tag{2.3}$$

we then obtain

$$h(x,t) = -\frac{(1 - P(T > t)]'\delta t}{P(T > t)\delta t}$$

$$= -\frac{d}{dt} \ln P(T > t) \tag{2.4}$$

Intergrating h between 0 and t, we obtain

$$P(T > t) = \exp(-\int_0^t h(x,t')dt') \tag{2.5}$$

If we want to justify the expression of the simple example model stated in the beginning, we make the assumption that $h(x,t)$ is constant. So the integral $\int_0^t h(x,t')dt'$ equals ht and expression 2.1 is satisfied.

We can introduce here an important simplifying assumption due to Cox, called proportional hazards. This assumes that the hazard function can be factored into a product of a function of the prognostic factors and a function of time in the form

$$h(x,t) = h_0(t)\psi(x) \tag{2.6}$$

where $h_0(t)$ is called the baseline hazard rate and $\psi(x)$ denotes the hazard rate of the patient. This factorisation assumes that there is a common hazard rate curve which is scaled by the prognostic factors.

Formally, the baseline hazard $h_0(t)$ is the hazard at time t of an individual whose x's, prognostic factors, are all zero. Usually, $h_0(t)$ is of little interest in itself, since it may depend on the prognostic factors. Thus, the Cox model assumes that the hazards of any two patients are proportional over time, ie., the ratio between the hazards is the same at any time t. This does not mean, however, that the hazard will not change over time. However, the Cox model assumes that changes in the hazard of any patient over time will always be proportional to changes in the hazard of any other patient and to changes in the underlying hazard over time. In other words, it is assumed here that there is a common hazard rate curve which is scaled by the prognostic factors.

To verify the previous assumption, let us consider a statistical test mentioned by Lee [4], where a check is performed to determine whether the assumption is valid for each new application. If we rewrite the survival time function using the proportional hazards hypothesis, we have, ($P(x > t)$ is denoted as $S(t,x)$)

$$S(t,x) = \exp(-\int_0^t h(x,t')dt') \tag{2.7}$$

$$\exp(-\int_0^t h_0(t')\psi(x)dt') = \exp(-\psi(x)\int_0^t h_0(t')dt')) \tag{2.8}$$

since ψ does not depend on t', and since $\exp(ab) = \exp(a)^b$, we have

$$S(t,x) = S_0(t)^{\psi(x)} \tag{2.9}$$

where

$$S_0(t) = \exp(-\int_0^t h_0(t')dt') \tag{2.10}$$

Expression 2.9 is important, because it is the central idea behind Cox analysis. As $h_0(t)$ is a positive function, $S_0(t)$ has values between 0 and 1. Because $S(t,x) = S_0(t)^{\psi(x)}$ is a probability function, values of ψ lie between 0 and $+\infty$. So the constraint $\psi(x) > 0$ is always satisfied. We can also note here that since the survival has an exponential relation with ψ, this implies that the closer ψ is to zero the better the survival is (since $S(t,x) \to 1$ (equation 2.10)).

Determing ψ is the main goal of Cox regression; the clinician can then classify the patients into categories. He/she takes into account the patients ψ value and determines the patients profile, so that he/she can make better decisions. Also with the ψ index (which can be viewed as a hazard index or a prognostic index) we can determine the importance of the prognostic factors.

2.1.2 The Function ψ

In order to calculate the hazard function it is necessary to know ψ. The nature of ψ is determined a priori. An assumption made is that the function $\psi(x)$ depends, apart from the prognostic factors x, also on a vector β that characterises the prognostic factors. Therefore, we can write

$$\psi(x) = \prod_{j=1}^{i} \exp(\beta_i x_i) = \exp(\beta_1 x_1 + \beta_2 x_2 + ... + \beta_i x_i), \tag{2.11}$$

or

$$\psi(x_i) = \exp(\beta_j x_j), \tag{2.12}$$

where β is a vector that has one element characterising each prognostic factor.

Whatever the value of ψ, its parameters, i.e., the β vector, is optimised so that the model accurately describes the data. The method suggested by Cox is the maximum likelihood procedure.

2.2 Maximum Likelihood

The likelihood function to optimise the parameters of the Cox regression model is quite different from the type of likelihood function encountered when designing an error function of a multi-layer perceptron in a classification or in a regression problem (Bishop [3]). In Cox regression the likelihood function includes the probability that the parameters describe accurately a distribution of points drawn from a series of probability distributions, namely, the hazard functions. These probability distributions are of a similar form, but distinct. It is not from a combination of distributions that one can draw points. The points come from different distributions.

The definition of the likelihood function in survival data analysis is actually the probability that all the patients in the set die.

The likelihood functions varies based on the assumptions made for the data. In this project we make two assumptions concerning the data:

- there is continuity of time, i.e., that two or more patients do not die on the same day, and
- there is no missing data, i.e., no information is missing from the data set. Actually we took care of this problem simply by not including those patients with missing information in the data set.

In order to derive the likelihood, we need to consider the fact that the baseline hazard function is completely unknown, and we want to infer the β vector. We therefore suppose we have a training data set $D = \{x_n, t_n\}$ consisting of pairs of vectors x_n of prognostic factors together with survival times t_n, where $n = 1, ..., N$. In order to determine a suitable form for the likelihood function it is convenient to re-express the data in an equivalent form as $\{x_n, \tau_n, I_n\}$ where $\{\tau_n\}$ denotes the survival times and $\{I_n\}$ represents labels which denote the identity of the patient who died at the corresponding survival time $\{\tau_n\}$. The joint probability of survival times and labels, given the prognostic factors and the parameters β, is

$$P(\{\tau_n, I_n\}|\{x_n\}, \beta) = P(\{I_n\}|\{\tau_n, x_n\}, \beta)P(\{\tau_n\}|\{x_n\}, \beta) \tag{2.13}$$

To determine the vector β we assume that $P(\{\tau_n\}|\{x_n\}, \beta)$ is independent of β.

If we suppose that the examples are ordered so that $\tau_1 < \tau_2 < .... < \tau_N$, then we can use the rules of probability to give

$$P(\{I_n\}|\{\tau_n, x_n\}, \beta) = \prod_{n=1}^{N} P(I_n, |I_1, ...I_n - 1, \{\tau_n, x_n\}, \beta) \tag{2.14}$$

2.2.1 The Risk set

For each failure time a risk set is defined. The risk set, denoted as $\Re_i$, is the set of patients m for which $t_m > t_n$. In other words, it is the set of patients alive at the time that patient n dies.

Let $\tau_1 < \tau_2 < ... < \tau_n$ be the ordered survival times of n patients and let f_j denote the patient that dies at time τ_j. Then, the definition of the risk set is

$$\Re(\tau_j) = \{i : t_i \leq \tau_j\}, \tag{2.15}$$

and in addition,

$$f_j = i \qquad iff \qquad t_i = \tau_j \Re(\tau_j)$$

. Suppose we have the survival times of four patients as shown in figure 2.1.

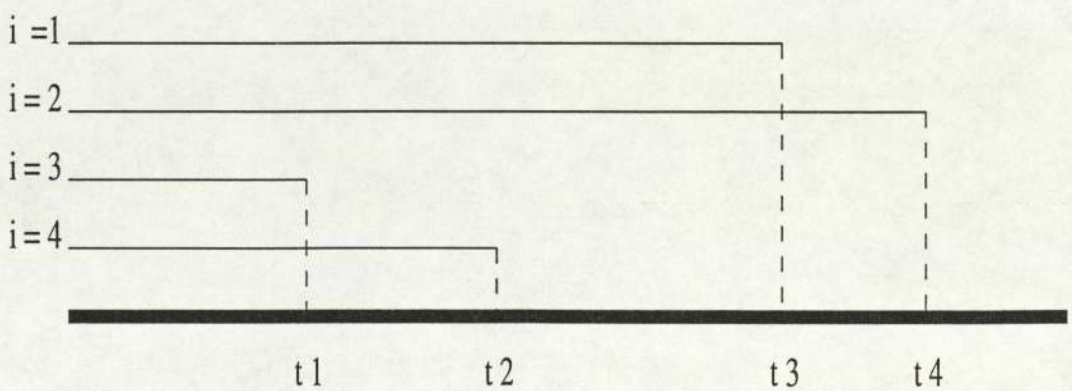


Figure 2.1: Example of survival times of four patients

The risk sets are:

- $\Re_1 = \{1, 2, 3, 4\}$
- $\Re_2 = \{1, 2, 4\}$

- $\Re_3 = \{1, 2\}$
- $\Re_4 = \{2\}$

This example shows clearly that the risk set $\Re_i$ at time t_i is the set of patients who are still alive at the moment another patient dies.

2.2.2 Maximum Likelihood derivation

Since the risk set is now defined we can derive the likelihood function. The probability of death of patient n in the time interval $t \leq T \leq t + \delta t$ is given by $h(x_n, t)\delta t$. So, given that one of the patients in the risk set dies in this time interval, the probability that it is patient n is given by

$$\frac{h(x_n, t)\delta t}{\sum_{m \in \Re_n} h(x_m, t)\delta t} = \frac{\psi(x_n)}{\sum_{m \in \Re_n} \psi(x_m)} \tag{2.16}$$

where equation 2.6 was used. From equations 2.14 and 2.16 we obtain the likelihood function which is of the form

$$\mathcal{L} = \prod_{n=1}^{N} \frac{\psi(x_n)}{\sum_{m \in \Re_n} \psi(x_m)} \tag{2.17}$$

So the final likelihood function is the product over all the patients in the set, of the hazard function of the patient n, divided by the sum of the hazard functions of the patients who belong in the risk set $\Re_n$. Here we see that the proportional hazards assumption has simplified the formalism since the baseline hazard function $h_0(t)$ does not enter the likelihood. In this manner, the likelihood function is obtained. It is a product rather than a probability, because terms that determine which individuals should be censored from among the survivors of each risk set have been omitted.

The log-likelihood is given by the expression

$$\mathcal{L} = \sum_{i=1}^{N} [(\beta . x_i) - \ln \sum_{k \in \Re_i} \exp(\beta . x_k)] \tag{2.18}$$

Since in standard Cox regression we need to obtain the β vector, we need to compute first the gradient of the log-likelihood with respect to β.

$$\frac{\partial \mathcal{L}}{\partial \beta} = \sum_{i=1}^{N} x_i - \sum_{i=1}^{N} \frac{\sum_{k \in \Re_i} (\psi_k . x_k)}{\sum_{k \in \Re_i} \psi_k} \tag{2.19}$$

Knowing the log-likelihood expression and its gradient according to β, we can obtain the β vector. We can use a standard algorithm such as scaled conjugate gradient to find the optimise β. This implementation is described in the next chapter.

2.2.3 Problem with the likelihood function

The problem faced in reality is that, in the case of the CRC data the first assumption made to derive the likelihood function does not hold for the cases of all patients. If time was measured with infinite precision, time could be considered as continuous. However, in survival data analysis, time is usually measured in days. Therefore it is common that more than one patients die on the same day.

The simplest computation method derived to include this possibility in the likelihood expression is given by Cox [1]:

$$P(e_1, ..., e_d) = \frac{(\prod_{i=1}^{d} \psi_i)}{\sum_{k \in \Re_i} \psi(k))^d} \tag{2.20}$$

where d is the number of patients who die on the same day.

As mentioned by Cox and Oakes, this expression is valid only if d is a small number compared to the number of individuals in the set. This approximation is used by most statistical packages for survival data analysis, and is also used by BMDP which is the statistical package used by the statisticians in the CRC trials in Birmingham University [16].

Chapter 3

Implementation of Standard Cox Regression

The standard Cox method is first tested on synthetic data, in order to be able to determine whether the implementation is correct, and then it was applied to the real data. These procedures are described in this chapter.

3.1 Implementation of toy problem 1

The objective of this implementation is to obtain the β vector using standard Cox regression. However, to be able to determine if our implementation is successful, we applied the method first on synthetic data, and we called it toy problem 1.

The methodology was as follows:

- First, we created the synthetic data using a random number generator and we chose it to be of d dimensions and length N. This indicates that the input data includes N patients, and that we know d prognostic factors for each of them. The synthetic data is derived from a normal distribution with mean 0 and variance 1.

- Then, we chose β to be a vector β_{init}, again by using a random number generator, from a normal distribution.

- In order to generate the survival times for the synthetic data, ψ is computed. A vector r having length N is generated again from a normal distribution. The

survival times were then generated by the relation $-log(r)/\psi$. Once they are generated they are sorted, and the data set is sorted according to the survival times.

- We computed the log-likelihood and its gradient according to β, using equations 2.18 and 2.19 respectively.
- The β vector was initialised again to some random vector, by using a random number generator for normal distribution.
- In order to obtain β we need to optimise it, using a standard algorithm, and we used scaled conjugate gradient. We optimised the vector β, and it was named β_{fin}.
- We compared the optimised β_{fin} with β_{init}. If those two vectors are similar then it means that our implementation of the standard Cox regression model has been successful.

The actual program for which we tested the method considers β to be of three dimensions, and also considers four cases of inputs, in this case, the number of input points. It implements the method using 10, 100, 1000, and 10000 points.

Table 3.1 presents the results obtained by the implementation of the standard Cox regression on the toy problem. This implementation was performed five times keeping the same β_{init}. As observed from the table, the values of β_{fin} in all the 3 dimensions of the variable get closer and closer to β_{init} as the number of points increases. This was observed for all five implementations.

A comparison of the two vectors is shown graphically in figure 3.1. The relation presented here is

$$\beta_{fin} - \beta_{init}.$$

If β_{fin} is close to β_{init}, the above relation should be close to zero. As seen clearly from the figure, as the number of points increases, it gets closer to zero in all the 3 dimensions, so this verifies that the implementation is correct.

The next step is to apply standard Cox regression on the real data. This implementation is described in the next section.

	Dim. 1	Dim. 2	Dim. 3	# Points
Beta_init	- 0.6927	1.3227	- 0.3623	
Beta_fin 1	- 1.1256	1.3654	- 0.9736	10
	- 0.3803	0.1968	- 0.3980	
	- 1.0261	1.5369	- 0.6697	
	-1.3318	1.6567	- 0.6912	
	- 1.6446	2.2870	- 1.4668	
Beta_fin 2	- 0.8903	1.3668	- 0.4687	100
	- 0.9182	1.3908	- 0.1853	
	- 0.5094	1.2414	- 0.4904	
	- 0.7916	1.2443	- 0.4733	
	- 0.6695	1.1474	- 0.3040	
Beta_fin 3	- 0.6442	1.2660	- 0.3464	1000
	- 0.6285	1.2548	- 0.3610	
	- 0.7093	1.2323	- 0.3425	
	- 0.7252	1.2813	- 0.3460	
	- 0.7386	1.2998	- 0.3755	
Beta_fin 4	- 0.6937	1.3100	- 0.3489	10000
	- 0.7068	1.3277	- 0.3683	
	- 0.6974	1.3318	- 0.3628	
	- 0.6994	1.3445	- 0.3616	
	- 0.7165	1.3255	- 0.3541	

Table 3.1: The values of β_{fin} compared with β_{init} done for 5 cases, for the number of input points ranging 10 - 10000

3.2 Implementation of standard Cox model on the real data

We applied standard Cox regression on the real data in order to obtain the coefficients of the prognostic factors, which is the β vector. By knowing the value of their coefficients we can tell their order of importance.

The step-up procedure done by the statisticians at the CRC trials units has selected 5 prognostic variables. So that

$$\psi = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5.$$

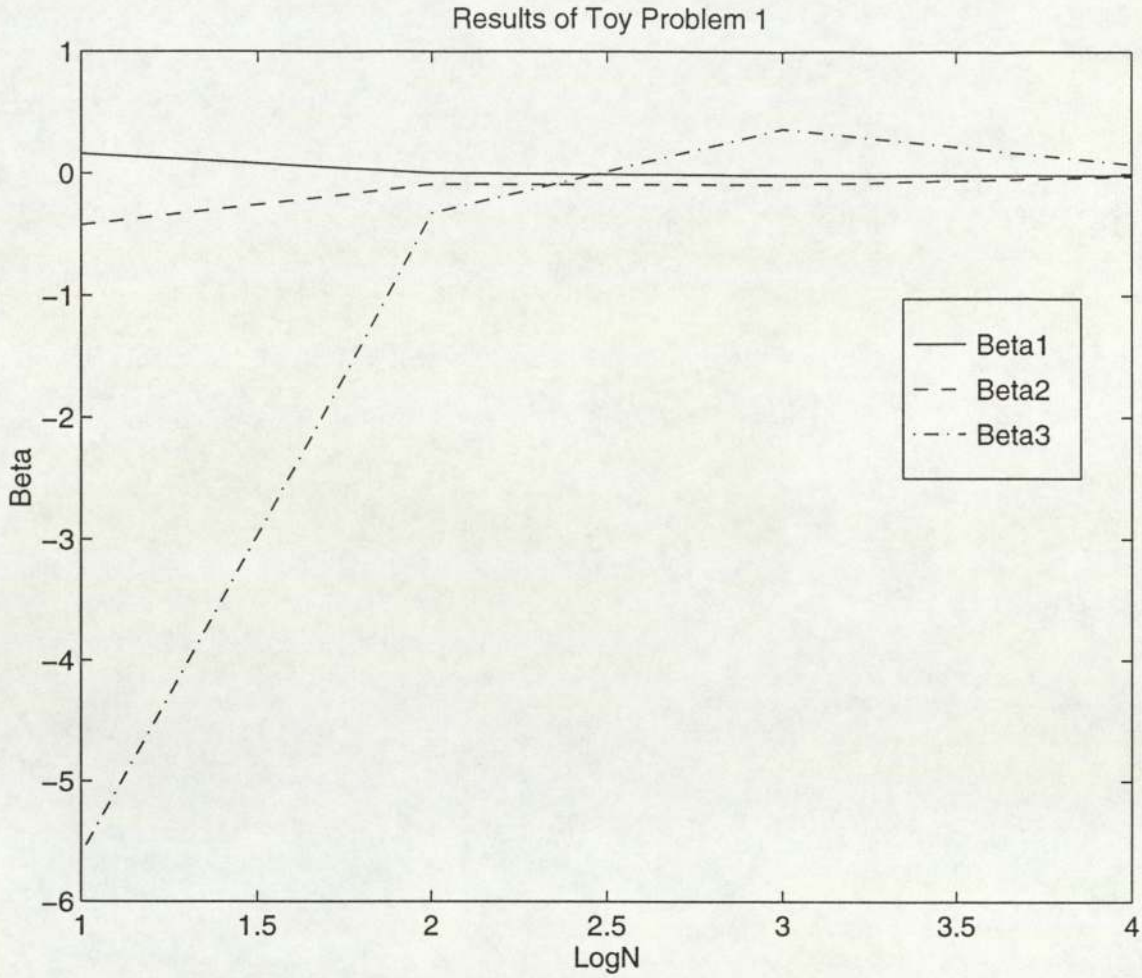


Figure 3.1: Graphical representation of the relation $\beta_{fin} - \beta_{init}$

These 5 prognostic factors are:

- Treatment
- Age
- Primary Tumour
- Dexamethasone
- WHO index

They are considered to be the most important prognostic factors, and the most essential ones for the analysis.

The coefficients obtained are given in the first half of table 3.2, and the prognostic factors are presented in order of importance according to their coefficients absolute value.

OUR RESULTS		STATISTICIANS RESULTS	
Coefficient	Value	Coefficient	Value
WHO	0.380175	WHO	0.3990
Primary Tumour	0.246429	Primary Tumour	0.3248
Dexamethasone	0.212985	Dexamethasone	0.2827
Treatment	- 0.106477	Treatment	- 0.2052
Age	0.065085	Age	0.1985

Table 3.2: Coefficients of the prognostic factors using standard Cox model as found by our implementation and as found in the CRC trials units

The log-likelihood was also calculated using the data for the 475 patients available and is:

$$\mathcal{L} = 2443.6$$

These results that we found by implementing standard Cox regression are comparable with the ones that the statisticians at the CRC trials units obtained. They have used a statistical package called BMDP and their results are presented in the second half of table 3.2. Also, the log-likelihood value that they obtained is very similar to the one found using our implementation:

$$\mathcal{L} = 2432.69$$

As one can see from the comparison of the two halves in table 3.2, the order of importance of the prognostic factors is the same, as is the sign of the values of their coefficients. The values themselves have a small difference, but this is because a different software was used in the two cases.

In conclusion, we can say that the implementation of the standard Cox regression is successful since the results obtained are comparable with the ones found in the CRC trials unit in Birmingham University.

Chapter 4

Cox Regression Using Neural Networks

In this chapter a new approach is introduced, which is Cox regression using neural networks. It was first tested on synthetic data and then on the real data. In the last section of the chapter, the results found from the two approaches are discussed.

4.1 Overview of Cox regression using neural networks

The purpose of the new approach, Cox regression using neural networks, is to check the performance of the model when a neural network is introduced. The real advantage of introducing a neural network lies in its capability to model a large variety of functions (Bishop et al [7]).

In order to derive a more flexible model, a simple procedure is to introduce quadratic terms of the form $x_i x_j$ or x_i^2, where x is the vector of the prognostic factors.

$$\psi = \exp(\beta x + \sum_{ij} w_{ij} x_i x_j) \tag{4.1}$$

We can then consider a generalisation of the Cox formalism in which we choose

$$\psi(x) = \exp(y(x; w))$$

Maximum likelihood is used to determine the values of the parameters in the network.

We suppose that our training data is the same as described in section 2.2. Following the same procedure as we did to derive the likelihood function for standard Cox regression, we derive the likelihood function for this model which is given by

$$\mathcal{L} = \prod_{i=1}^{N} \frac{\psi(x_i)}{\sum_{j \in \Re_i} \psi(x_j)} \tag{4.3}$$

It can also be written as

$$\mathcal{L} = \prod_{i=1}^{N} \frac{\exp(y(x_i; w))}{\sum_{j \in \Re_i} \exp(y(x_j; w))}. \tag{4.4}$$

To be able to work with the likelihood expression we take the log-likelihood which is:

$$ln\mathcal{L} = \sum_{i=1}^{N} [\ln(\psi(x_i)) - \ln \sum_{j \in \Re_i} \psi(x_j)] \tag{4.5}$$

Combining equations 4.2 and 4.5 we get

$$ln\mathcal{L} = \sum_{i=1}^{N} [y(x_i; w) - \ln \sum_{j \in \Re_i} \exp(y(x_j; w))] \tag{4.6}$$

We can determine the parameters w of the neural network by maximising $ln\mathcal{L}$ with respect to w using a standard algorithm such as scaled conjugate gradient. To achieve this we need to find the gradient of the log-likelihood with respect to w. In practice though, we consider the negative logarithm of the likelihood and by minimising it with respect to w, we obtain the optimised parameters of the network.

In order to derive the gradient of the likelihood with respect to the parameters of the network w, we apply the chain rule. So we have the expression:

$$\frac{\partial \mathcal{L}}{\partial w} = \sum_{i=1}^{N} \frac{\partial \ell}{\partial \psi(x)} \frac{\partial \psi(x)}{\partial w} \tag{4.7}$$

We can easily calculate the partial derivative of the likelihood with respect to $\psi(x)$ from the log-likelihood expression 4.5

$$\frac{\partial \mathcal{L}}{\partial \psi(x_i)} = \sum_{i=1}^{N} [\frac{1}{\psi(x_i)} - \frac{1}{\sum_{j \in \Re_n} \psi(x_j)}]. \tag{4.8}$$

We can also find the partial derivative of ψ with respect to w from equation 4.2

$$\frac{\partial \psi(x_i)}{\partial w} = \psi(x_i)\frac{\partial y_i}{\partial w}. \tag{4.9}$$

Finally, by substituting equations 4.8 and 4.9 into the chain rule equation 4.7, the gradient of the log-likelihood with respect to w is calculated as

$$\frac{\partial \mathcal{L}}{\partial w} = \sum_{i=1}^{N}[1 - \frac{\psi(x_i)}{\sum_{j\in\Re_i}\psi(x_j)}]\frac{\partial y}{\partial w} \tag{4.10}$$

which can also be written as

$$\frac{\partial \mathcal{L}}{\partial w} = \sum_{i=1}^{N}[1 - \frac{\exp(y(x_i;w))}{\sum_{j\in\Re_i}\exp(y(x_j;w))}]\frac{\partial y_i}{\partial w}. \tag{4.11}$$

4.2 Implementation of toy problem 2

As before, we want to test the performance of the present method. We therefore apply it to synthetic data first, in order to test its performance.

The procedure is similar to toy problem 1, which was described in section 3.1:

- We generate the synthetic data using a random number generator, but this time having one dimension. As before, the synthetic data generated are normally distributed.
- We choose y_{init} to be equal to some function of the prognostic factors x, $f(x)$.
- We compute the likelihood and its gradient in terms of w, so that we can determine the parameters of the network.
- We then initialise the parameters of the network w by constructing the two layer feed-forward neural network.
- The weights are then optimised by using a standard technique such as scaled conjugate gradient.
- The output of the network y_{fin} is then calculated.

In order for our implementation to be successful, we expect the output of the neural network y_{fin} to be close to the initial function stated, namely y_{init}.

Several functions have been tested, starting from the most simple ones, such as $\sin x$, to complicated functions, such as $\sin x^2 + \cos x^4$.

4.2.1 Implementation of simple functions

To start with, we used a simple function

$$y = \sin x$$

As one can see from figure 4.1, in this very simple example, the prediction of the network gets closer to the original values one as the number of points increases. When using 10 points, presented in the first graph, the prediction is far from accurate; it is actually not even close to the original function. With 100 points, as shown in the second graph, the prediction is again quite bad, but certainly much better than in the case of 10 input points. When having 1000 points, which is the third case, we see that the prediction of the network is quite good as the shape of the two graphs is the same, but the predicted one is a bit lower than the original. For the case of 10000 input points which is the fourth case, it is clearly observed that the prediction of the neural network is most accurate. Here, it seems that the network was able to determine the right weights to use, so as to predict the expected function.

Taking into account figure 4.2, we can see some improvements regarding the performance of the network. That is because the results are when using the same initial function (same initial hazard function) $\sin x$ but using 3 hidden neurons. We also have four cases of the number of input points 10, 100, 1000 and 10000. The important point here is that comparing the two figures, figure 4.1 and figure 4.2, one sees clearly that the network has better performance not only when the number of inputs increase, but also when the number of hidden neurons increase as well. This justifies the fact that even when having 100 input points when using the 3 hidden neurons, the network can identify the initial hazard function quite accurately. Of course, in the case of 1000 input points the prediction is much better and at 10000 input points we have the most accurate prediction of all the cases. In the last case the network can identify the hazard

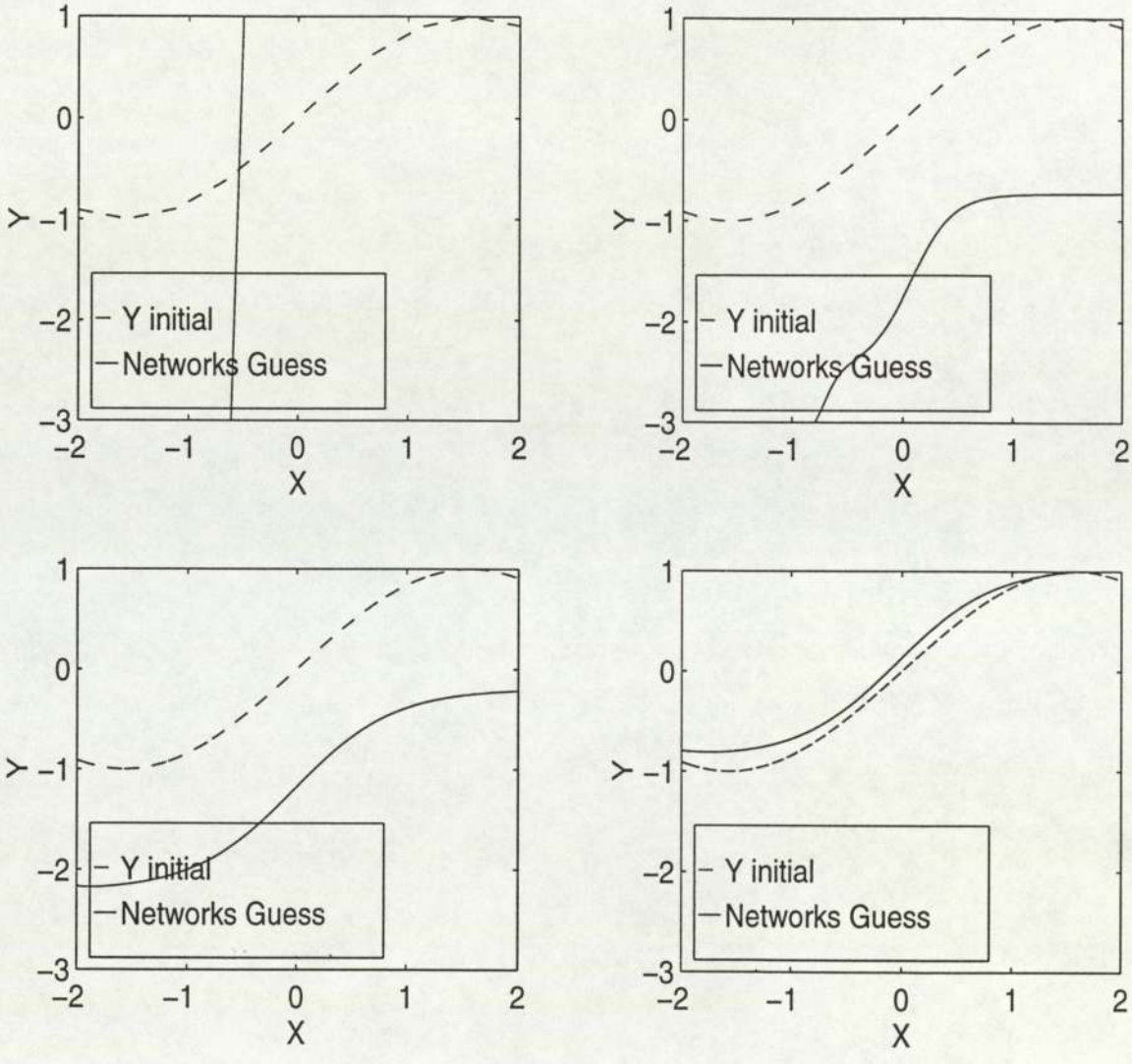


Figure 4.1: Function $\sin x$ using 1 hidden neuron and input points ranging from 10-10000. The first figure has 10 input points, the second 100, third 1000 and fourth 10000

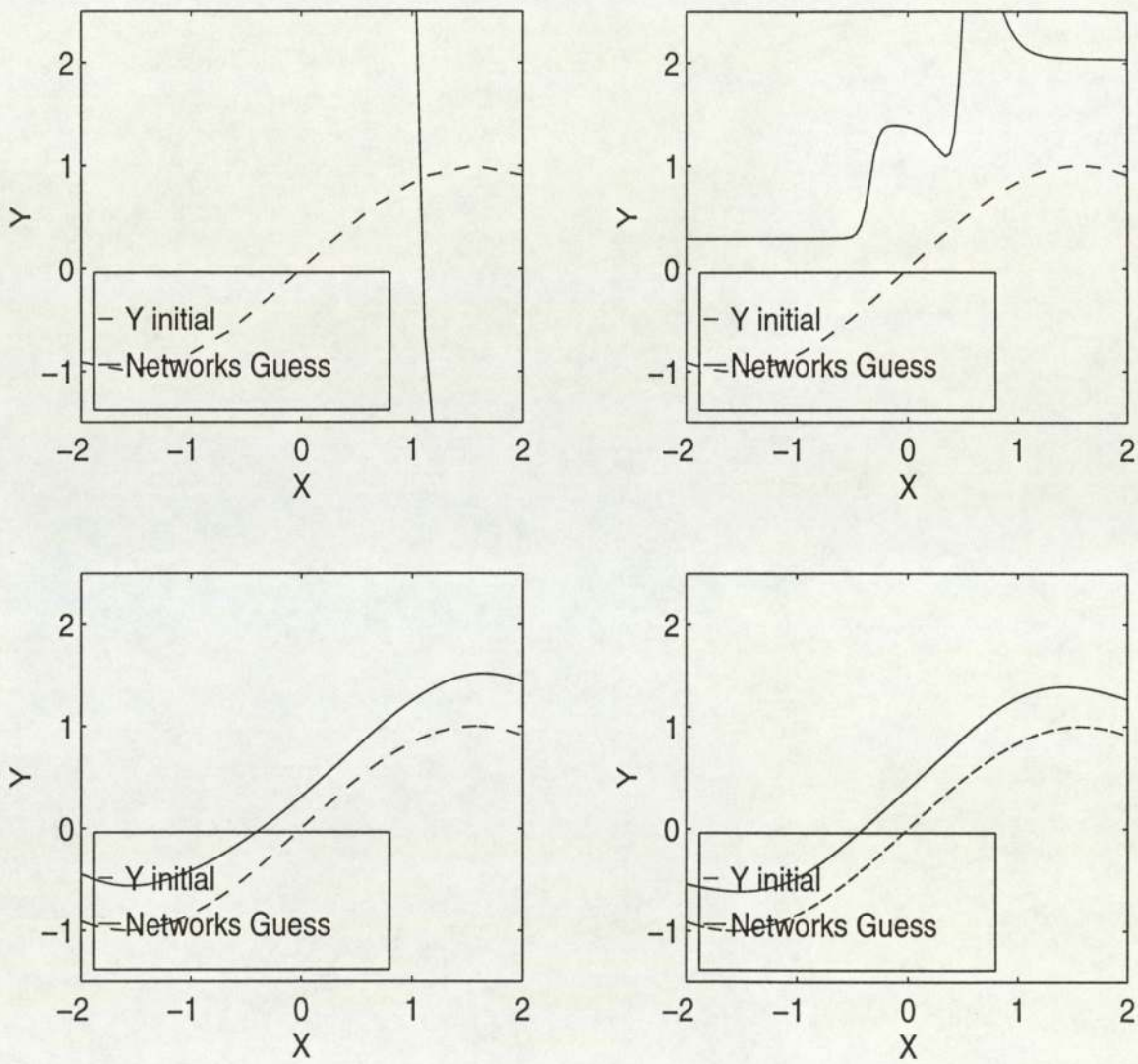


Figure 4.2: Function $\sin x$ using 3 hidden neurons. The first figure has 10 input points, the second 100, third 1000 and fourth 10000

function most accurately. In the case of 100 input points it can guess the shape of the function well, but is a bit lower than it. In the case of 1000 points it again guesses the shape of the function but now is above it. When using 10000 input points, which is the last case, the network predicts the function most accurately. There is only a small difference between the two curves, a small offset.

4.2.2 Implementation of more difficult functions

Lets consider now a more difficult function, which is

$$y = \cos(x^2 + 3x)$$

Figure 4.3 has this implementation. It was again tested using 10, 100, 1000 and 10000 input points. In this case, we started with using 3 hidden neurons since the hazard function we need to predict is a more difficult one than the previous one. It is clearly seen from this figure that the performance of the network gets better and better as the number of inputs increases.

Figure 4.4 represents the results of the training of a network that has 4 hidden neurons and takes 1000 points as inputs. The network in this case is trained for 400 iterations. From this figure one observes that even with 1000 input points, the network can predict the shape of the survival function accurately. Comparing figures 4.3 and 4.4, we can conclude that increasing the number of hidden neurons leads to better network performance.

We then tried to train the network using a greater number of iterations while keeping the number of hidden neurons and input points the same as before. This enables us to check if the performance of the model is affected. Figure 4.5 represents this case. Here, the network has 4 hidden neurons, with 1000 input points and it was trained for 800 iterations. As seen from the figure, the prediction of the network is slightly better than the one in figure 4.4. We can say, therefore, that the prediction of the network depends on the number of iterations that the network is trained for.

In figure 4.6 the network was trained using 1000 input points as before, but this time it was constructed with 5 hidden neurons. It was again trained for 400 iterations, and as can be seen, it has a better performance compared with the results in figure

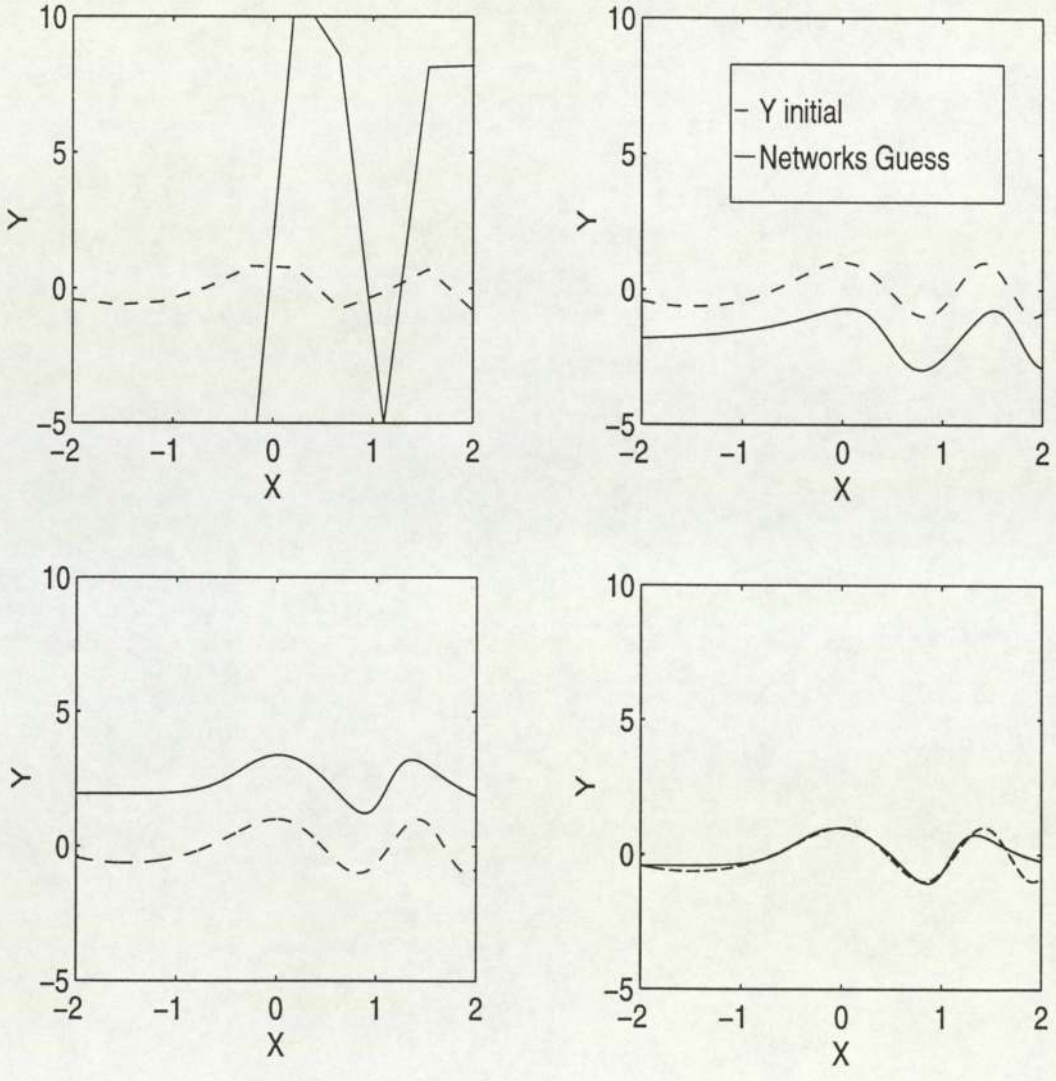


Figure 4.3: Function $\cos(x^2+3x)$ using 3 hidden neurons. The first figure has 10 input points, the second 100, third 1000 and fourth 10000

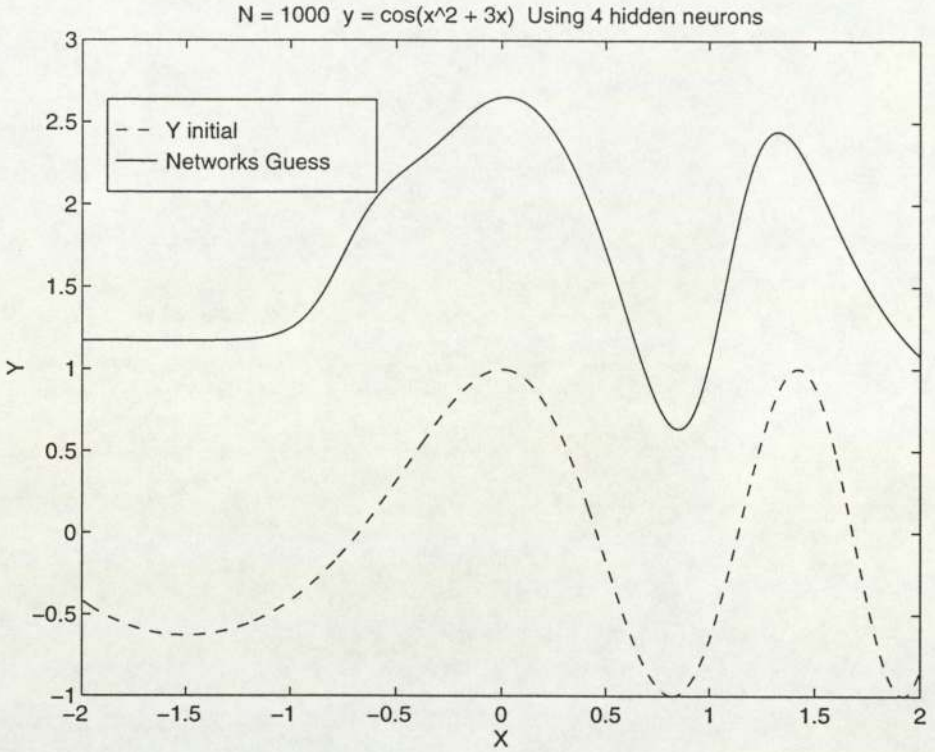


Figure 4.4: Function $\cos(x^2+3x)$ using 4 hidden neurons and 1000 input points, trained for 400 iterations

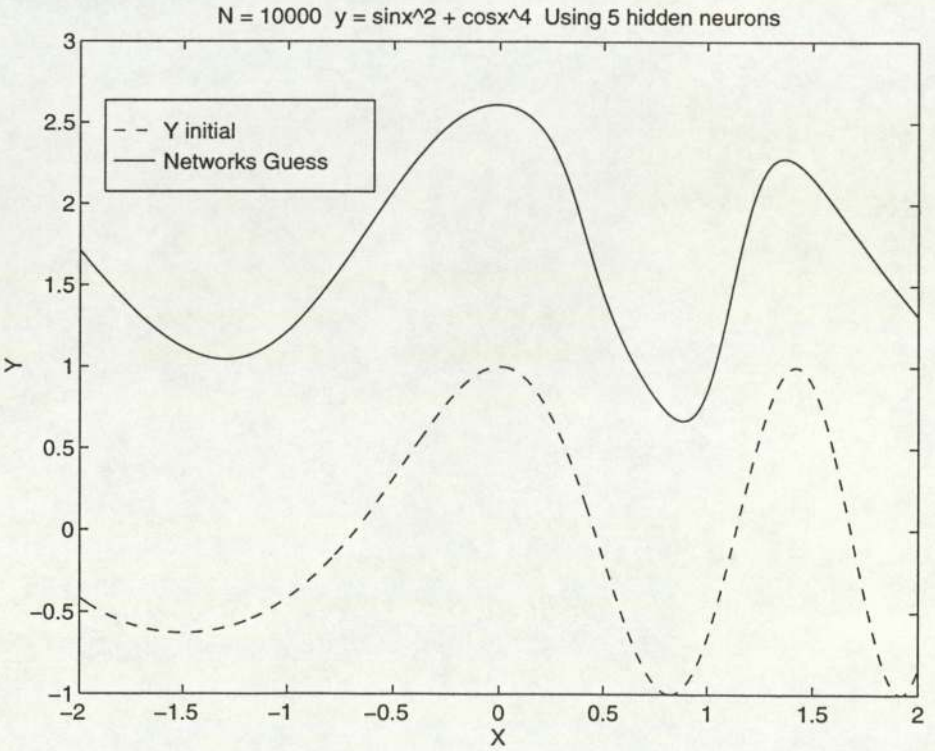


Figure 4.5: Function $\cos(x^2+3x)$ using 4 hidden neurons and 1000 input points, trained for 800 iterations

4.4. We again notice from these two figures that, as the network has an increasing number of hidden units, its performance improves. This indicates once again that our implementation of the new model, Cox regression using neural networks, seems so far to be correct.

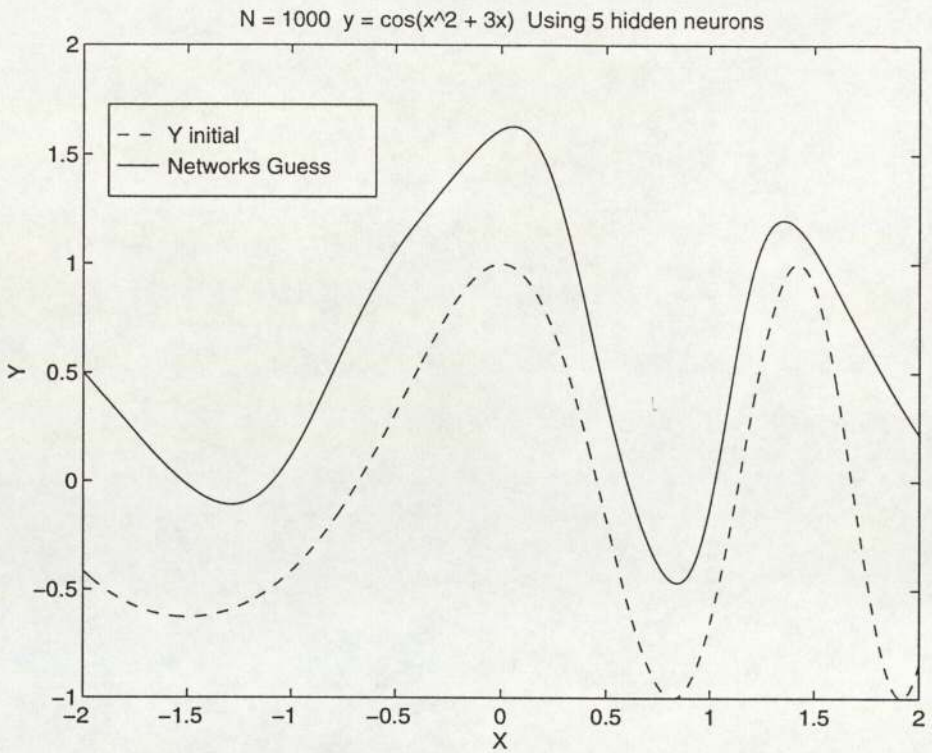


Figure 4.6: Function $\cos(x^2+3x)$ using 5 hidden neurons and 1000 input points, trained for 400 iterations

In figure 4.7 we used the same network construction as in the one in figure 4.6, but this time it receives 10000 input points. The difference between the two figures is clear enough. The network in figure 4.7 could predict the hazard function quite well, much better than in the case of figure 4.6. This suggests that the more input points we give to the network, the better the prediction it can make.

In figure 4.8 the function was implemented using 10 hidden units. Here we see that the estimation is not so good. That means that, as the number of hidden neurons keep increasing, the network performance degrades after a point. This is expected, because overly complex networks overfit the data, corresponding to high variance, and they have poor performance (Bishop [3]). That is explained better in section 4.3.1. Thus, in cases where there are more hidden neurons than needed, the network does not have a good performance, and that is only to be expected.

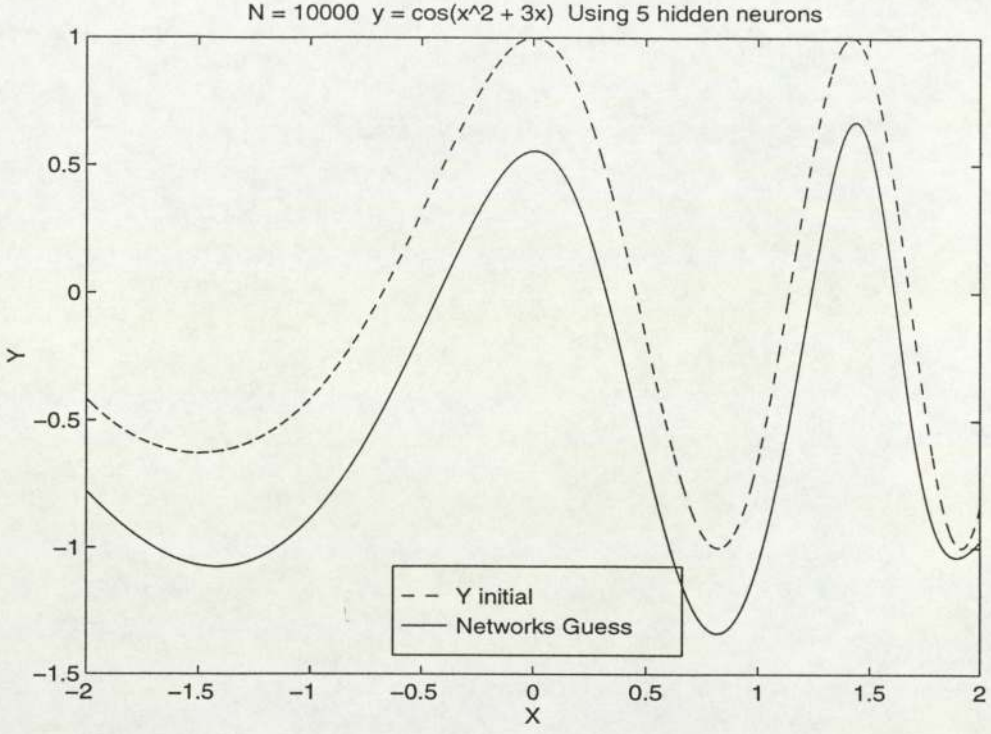


Figure 4.7: Function $\cos(x^2 + 3x)$ using 5 hidden neurons and 10000 input points, trained for 400 iterations

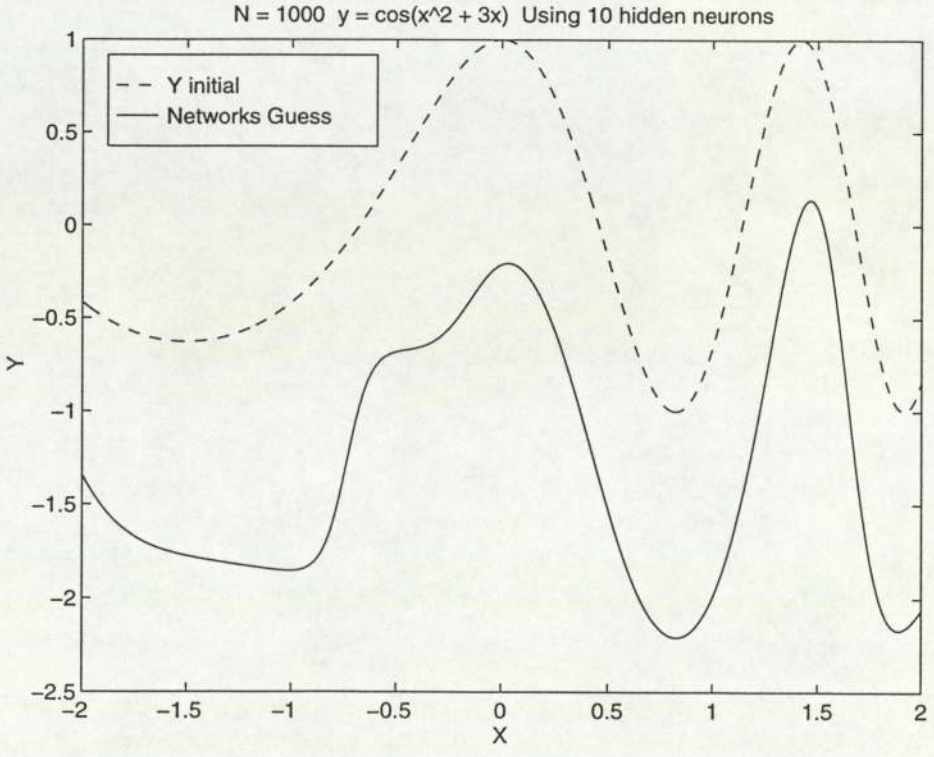


Figure 4.8: Function $\cos(x^2 + 3x)$ using 10 hidden neurons and 1000 input points, trained for 800 iterations

We then moved on and tried an even more difficult function for the neural network to predict, one that has more bumps. This survival function is:

$$y = \sin x^2 + \cos x^4$$

Figure 4.9 shows the results of a network that has 5 hidden neurons, and receives 10000 input points. It was trained for 400 iterations and, as one can see from the figure, the networks prediction is not so accurate. There is a big offset between the two curves. Knowing that we are dealing with a difficult graph to predict we can say that since the network could at least find the shape of the curve, this is quite an encouraging result. Most probably, in this case, the network should have been trained for more iterations, or have more hidden neurons.

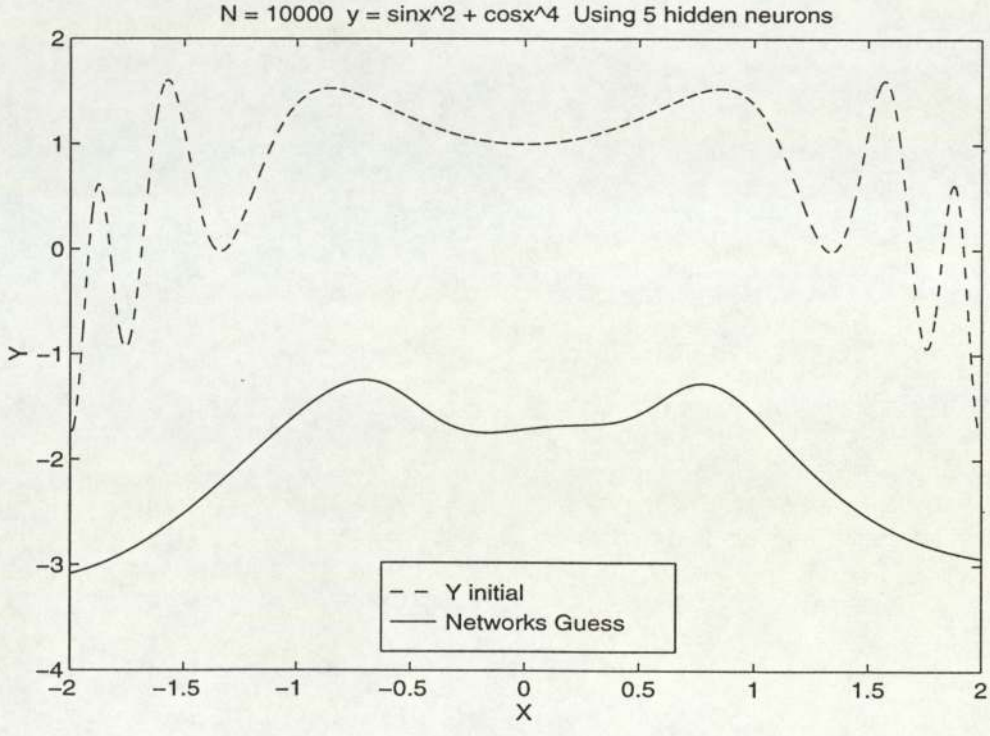


Figure 4.9: Function $\sin x^2 + \cos x^4$ using 5 hidden neurons and 10000 input points, trained for 200 iterations

Figures 4.10 to 4.12 represent the results when using the same survival function but with 1000 input points.

Figure 4.10 shows much better results than figure 4.9, and that shows once again that the network with more hidden neurons has a better performance. Here, the network has 9 hidden neurons, trained for 800 iterations. In the case of figure 4.11,

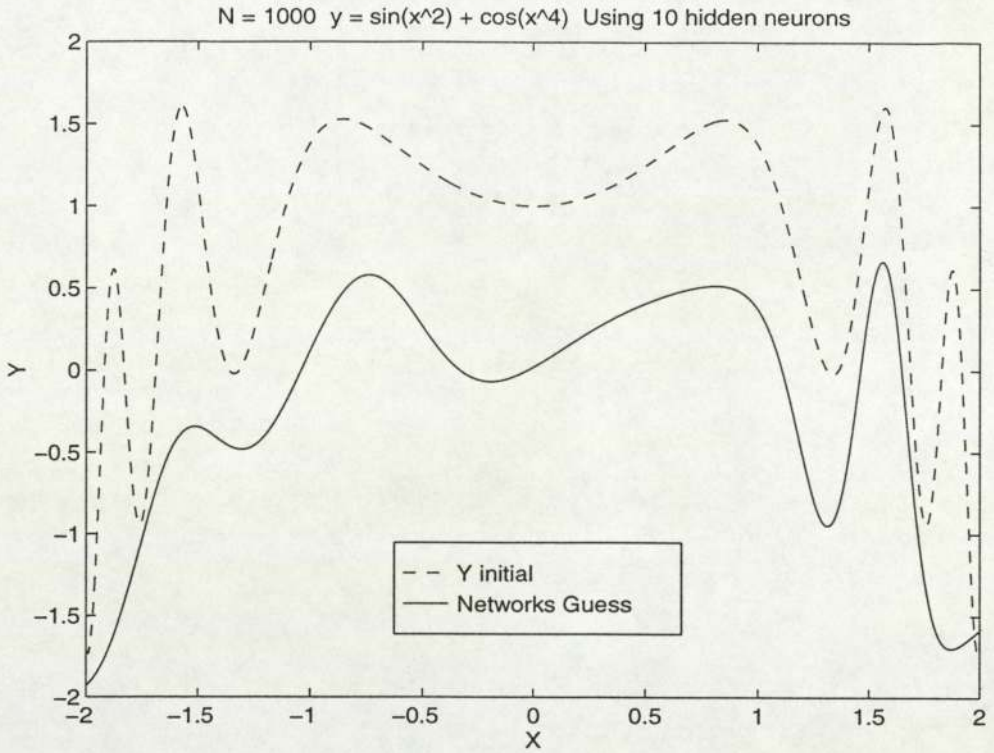


Figure 4.10: Function $\sin x^2 + \cos x^4$ using 9 hidden neurons and 1000 input points, trained for 800 iterations

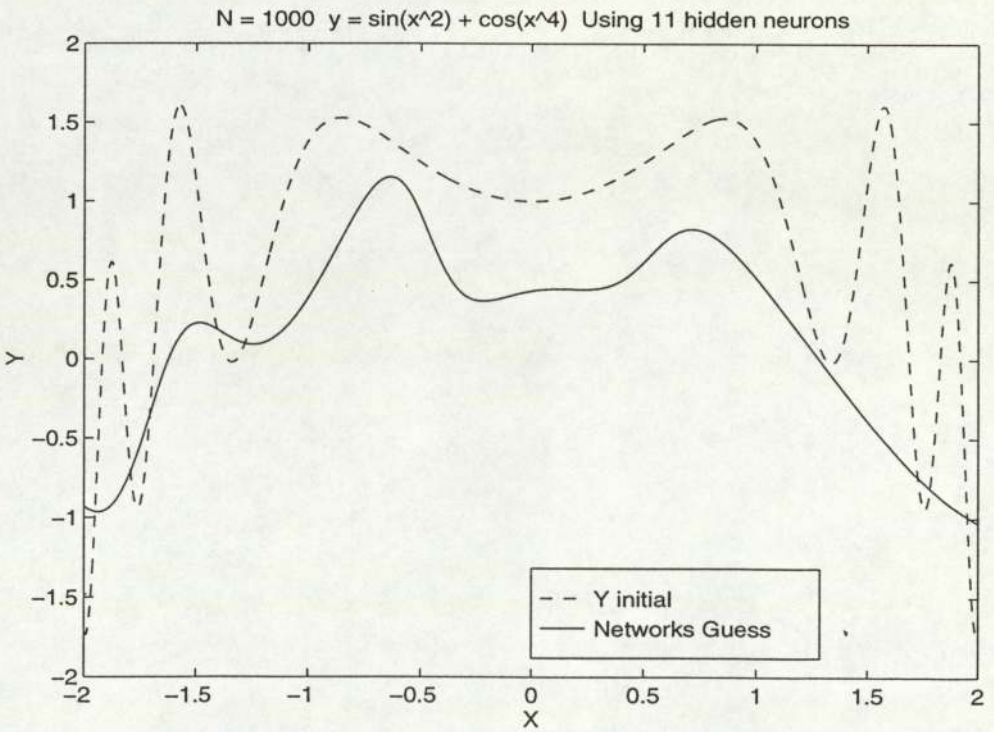


Figure 4.11: Function $\sin x^2 + \cos x^4$ using 11 hidden neurons and 1000 input points, trained for 800 iterations

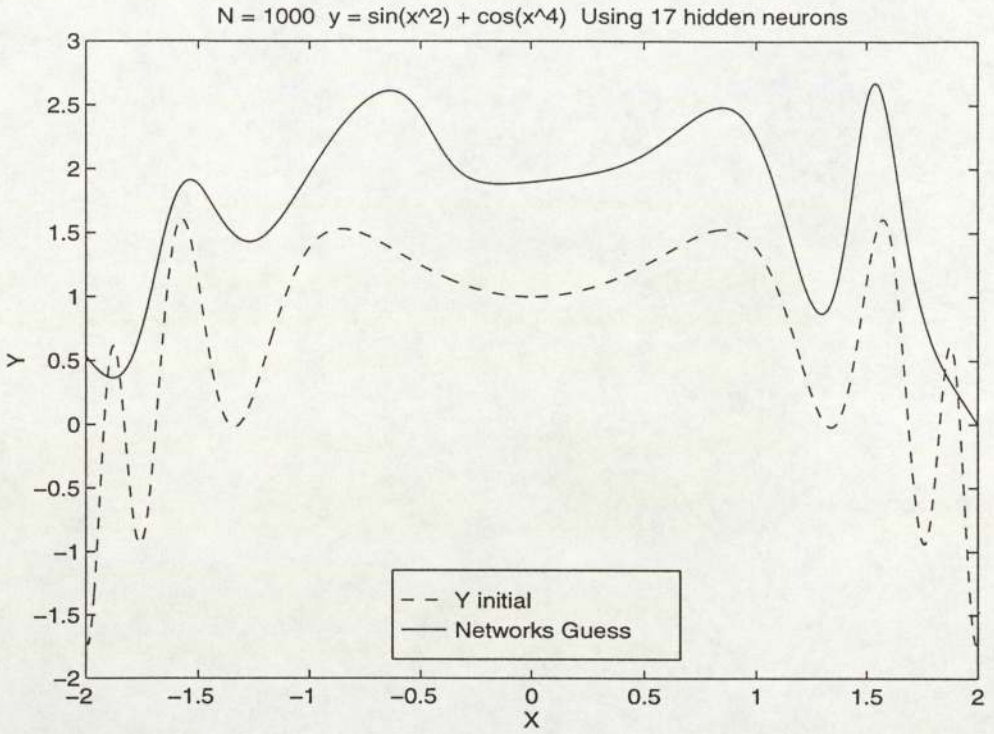


Figure 4.12: Function $\sin x^2 + \cos x^4$ using 17 hidden neurons and 1000 input points, trained for 800 iterations

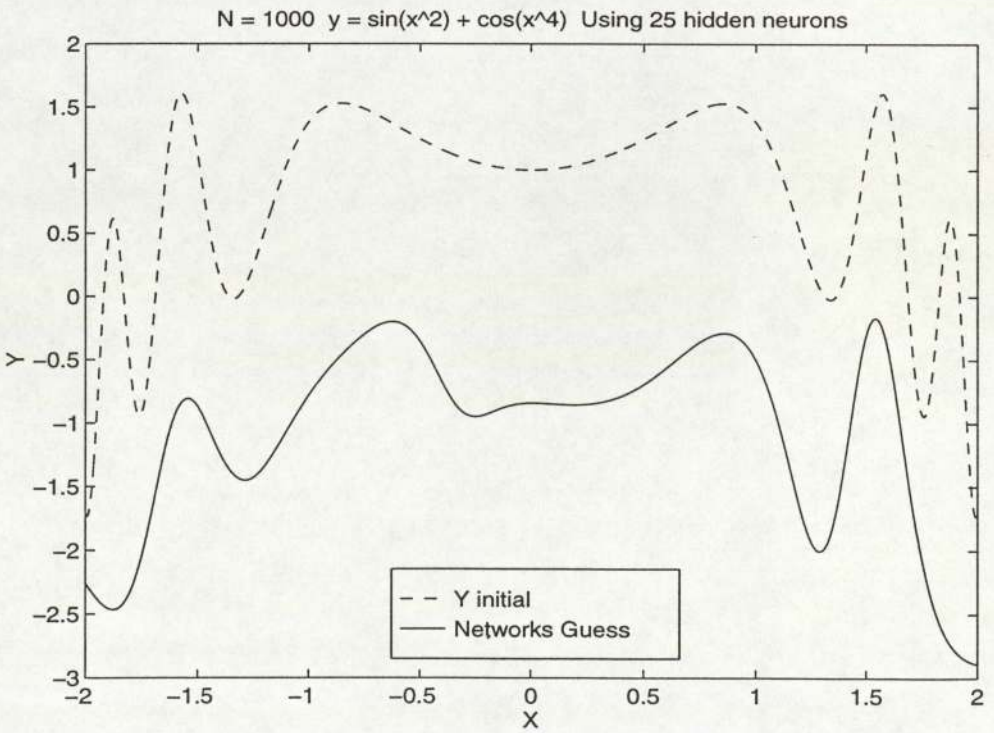


Figure 4.13: Function $\sin x^2 + \cos x^4$ using 25 hidden neurons and 1000 input points, trained for 800 iterations

the prediction of the network is even much better. In this case having 11 hidden neurons, and again trained for 800 iterations the network is able to predict fairly well this difficult hazard function. From figure 4.12 we can see that when using 17 hidden neurons and 800 iterations the network can predict better the hazard function. One can notice that the network can predict the curve more accurately, but the predicted curve is actually above the original one. Also, the offset of the two curves is the least from among all the cases up to now using this function.

From figure 4.13, we see as similar event as in figure 4.8, which is that after a point, when the number of hidden neurons increases, the network does not have a good performance. This observation is quite useful for us because it shows that the approach used here seems to be successful.

Observing all the results presented in this section, we gained confidence that the new approach, Cox regression with neural networks, is going to be correct. It seems that it was quite well implemented in the case of the toy problem, tested on synthetic data. Actually this was the whole idea of the toy problem, namely, to see if the method works successfully on synthetic data, which may imply that it will work as well on the cancer data that we have.

One can say, though, that the problem faced here is that we need a large data set to be able to have an accurate prediction of the network. That is because we are not actually fitting a line to some points, but we are trying to predict the hazard function of the patients (y_{init}) given their prognostic factors, and their survival times. The actual purpose of this toy problem is to verify that by predicting the hazard curve of the patients hazard, we will be able to tell his/her hazard value as and when a new patient comes along to the trial .

4.3 Implementation of the new approach on the real data

From the implementation of the approach on the toy problem we gained confidence that this is a successful method, and so we applied it on the real data. The purpose here is to find the values of the log-likelihood and compare them with those found using

the linear method, namely, standard Cox regression.

4.3.1 Structure of the network

The network used, as mentioned before, is a two layer perceptron. To have an idea of the complexity of the model that can be used regarding the number of the hidden neurons, we consider equation 4.12, where ϵ is an approximate rate of misclassification, W the number of the parameters in the network and $N_{pattern}$ the number of data points available (Bishop [3]).

$$N_{pattern} \simeq \frac{W}{\epsilon} \tag{4.12}$$

For $\epsilon = 0.1$ the minimum number of patterns necessary to train the network is about ten times the number of the parameters W. The network is a two layer perceptron with sigmoidal activation units. In this case, the number of parameters of the network are found from the equation

$$\overbrace{(no.\ inputs}^{1_{st}\ \text{layer}} + \overbrace{1}^{\text{bias } 1_{st}\ \text{layer}} + \overbrace{no.\ outputs)}^{2_{nd}\ \text{layer}} \times no.\ hidden\ units + \overbrace{no.\ outputs}^{\text{bias of } 2_{nd}\ \text{layer}}.$$

Using the previous equation we derive table 4.1, which shows the number of parameters for a network according to its number of hidden units. As seen from the table, a rough estimate shows that in the case of the amount of the current data, up to seven hidden neurons may be used. In our case though, as will be discussed in one of the next sections of the chapter, we trained the network on up to four hidden units due to the slow execution time of the programmes.

Number of hidden units	1	2	3	4	5	6	7
Number of parameters W	8	15	22	29	36	43	50

Table 4.1: The number of parameters for a neural network according to its number of hidden neurons

4.3.2 Cross validation procedure

In order to compare the performance of the linear model and the neural network, the data set is divided into a training and a test set. Since the data set is very small a cross validation procedure is performed in order to determine the generalisation performance of the network.

In practice, the availability of the labelled data may be very limited and we may not be able to afford the luxury of keeping aside part of the data set for model comparison purposes (Bishop [3]). As mentioned in section 1.2 the data set we are using consists of 475 patients, so in our case we need to use the cross-validation procedure. We divided the first 470 points of the data set into 10 equal sets of 47 points, which were used as the validation set. To evaluate the performance of the network we take each time a pair of a training set, which are the remaining 428 points, and a validation set. For each initial guess of the weights, we train the network using the training set and then test its performance, by evaluating the log-likelihood function using the validation set. For each couple of a training and validation set, we made five initial guesses for the weights. For each one of these guesses, the network parameters are optimised using the scaled conjugate gradient algorithm. This process was repeated for all the pairs of training and validation sets, and the log-likelihood results are averaged first over the five values obtained, and then over the ten values obtained from all the pairs. At the end, we had two values of the log-likelihood for the training and the validation set. The whole cross-validation procedure is explained in figure 4.14.

The cross-validation procedure was performed for networks with 1, 2, 3 and 4 hidden neurons. Figure 4.15 show the results of the cross-validation procedure. It presents the likelihood values of the training and the validation set in two different graphs so that one can see the difference. It also shows the error bars of the likelihood values. As seen from the figure, the value of the log-likelihood increases until the case of 2 hidden neurons and then it declines. In the case of the validation set, it increases as the number of hidden neurons increase.

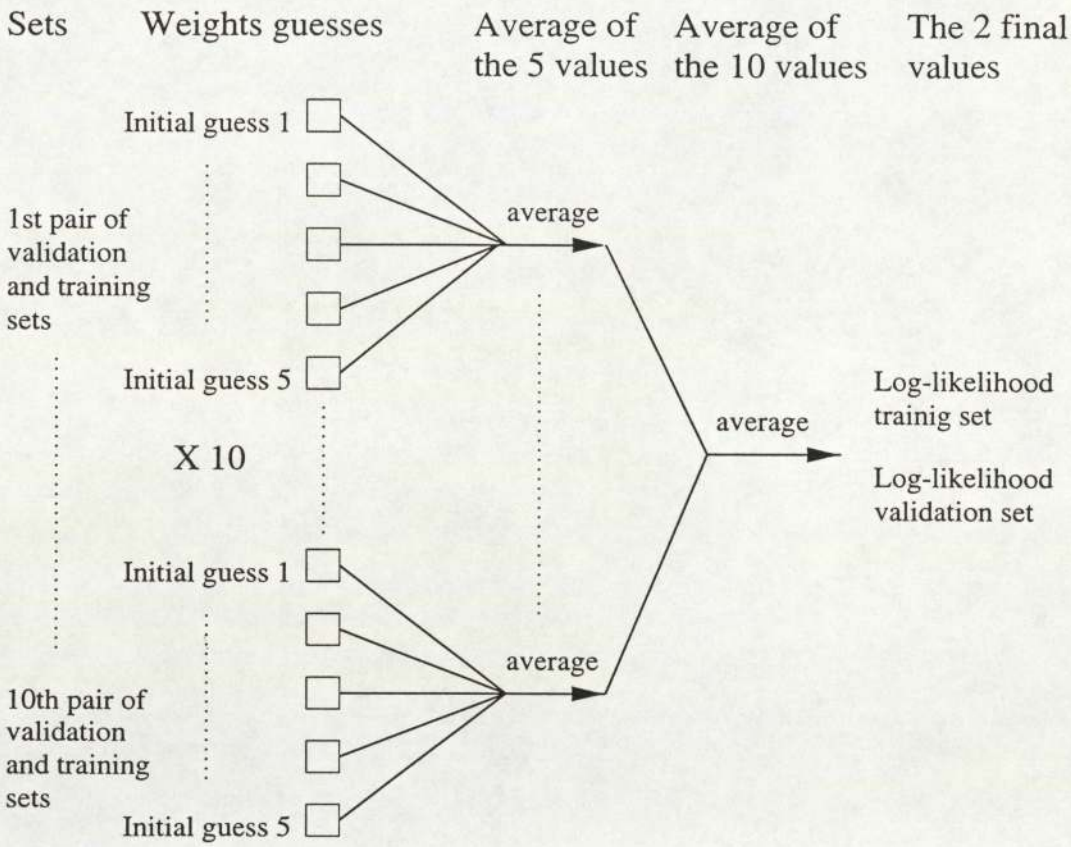


Figure 4.14: Description of the cross-validation procedure

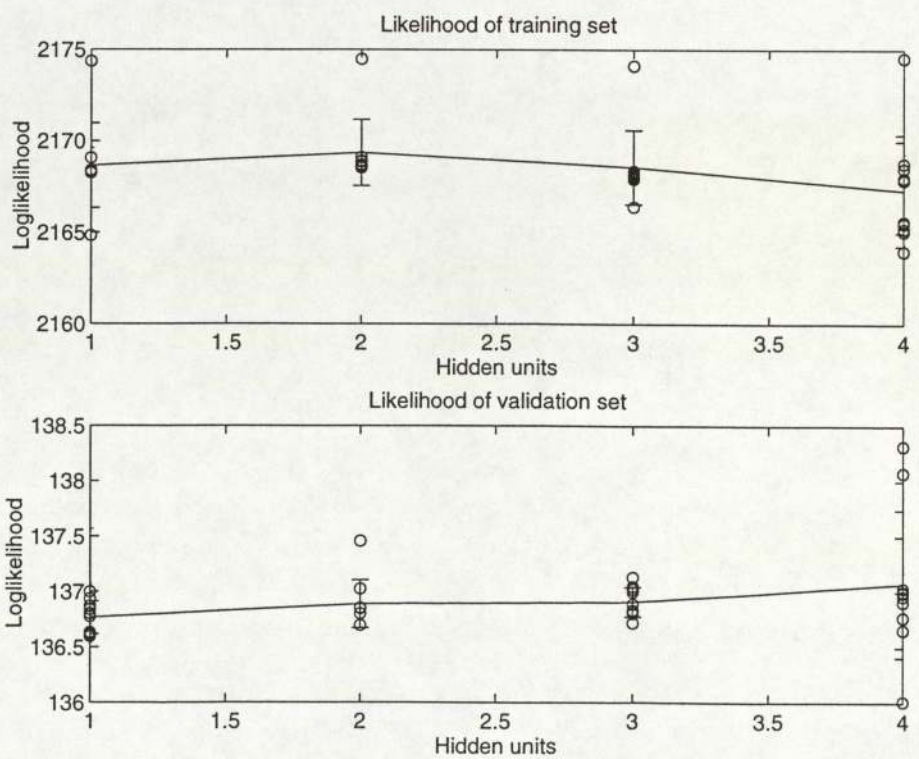


Figure 4.15: Log-likelihood values for training and validation sets

4.3.3 Comparison of the Cox and the neural network model

In order to compare the performance of the two models, we found the log-likelihood values of the training and validation pair sets using the Cox regression method as we did in the case of the cross validation procedure where we used the neural network method. To be able to see the comparison between these results figures 4.16 and 4.17 were created.

Figures 4.16 and 4.17 present the graphical representation of the difference between the log-likelihood values found when using Cox method and the ones using the neural network method. The results are found for both the training and the validation sets.

As seen from figure 4.16 the values of the difference between the log-likelihood values using the Cox model and the neural network model is below zero for the training set for all the cases of the hidden neurons. This implies that the log-likelihood value obtained by the neural network method is always greater than the one found by the standard Cox method.

In the case of figure 4.17, which shows the difference when using the validation set, one observes that the curve is declining and that it lies a bit above zero, starting from value 1.2. That means that the value of the log-likelihood when using the standard Cox model is a bit larger than the one using the neural network model. However the fact that the difference is not big and that it decreases is encouraging. For 1 to 2 hidden neurons the difference decreases steeply, for 2 to 3 decreases smoothly, and from 3 to 4 hidden neurons it again decreases steeply. This implies that the log-likelihood values for the validation set do not have much difference.

4.4 Log-likelihood results obtained

The data set was split into a training and a test set after performing the cross-validation procedure. We calculated the log-likelihood of the training and the test set, using 4 different constructions of networks, and also using the linear approach. The results are presented in table 4.2. In this case, we have used all the five prognostic factors that we used in section 3.2. To remind those prognostic factors, they are: WHO index, Primary Tumour, Dexamethasone, Treatment and Age.

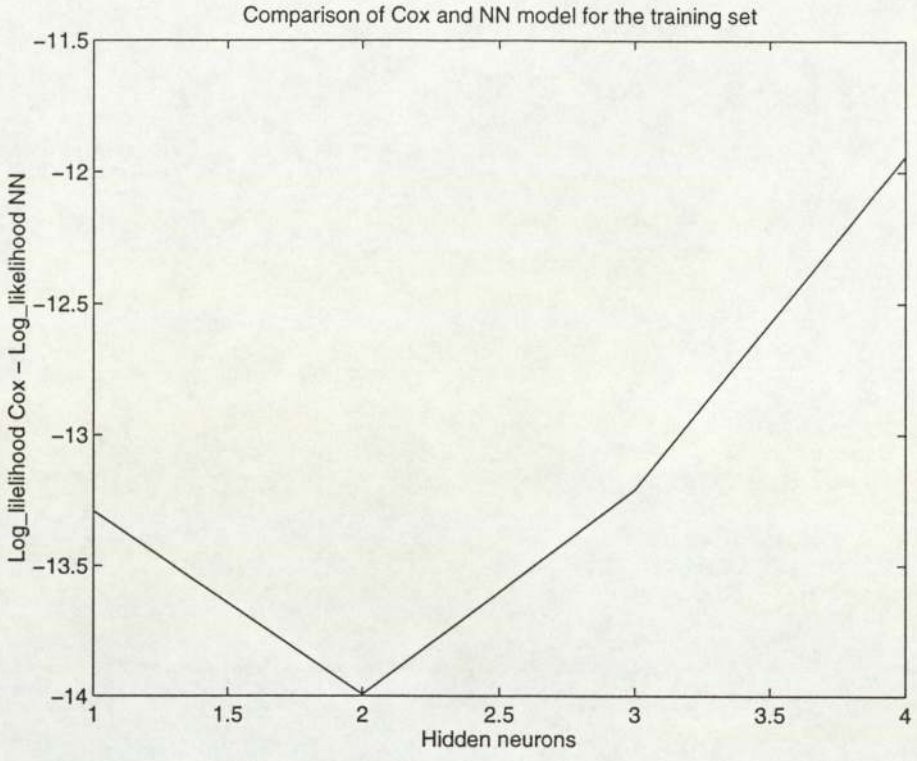


Figure 4.16: Comparison between Cox method and the neural network method for the training set. It actually represents graphically the relation: $Log\_likelihoodCox - Log\_likelihoodNN$

Parameters of the networks			- L training	- L test
Num. Inputs	Num. Hid. Un.	Num. Outputs		
5	1	1	2198.81	117.81
5	2	1	2199.58	117.77
5	3	1	2199.80	117.84
5	4	1	2189.80	117.69
Linear case			2185.72	118.56

Table 4.2: Log-likelihood results using 4 networks constructions, and using five prognostic factors

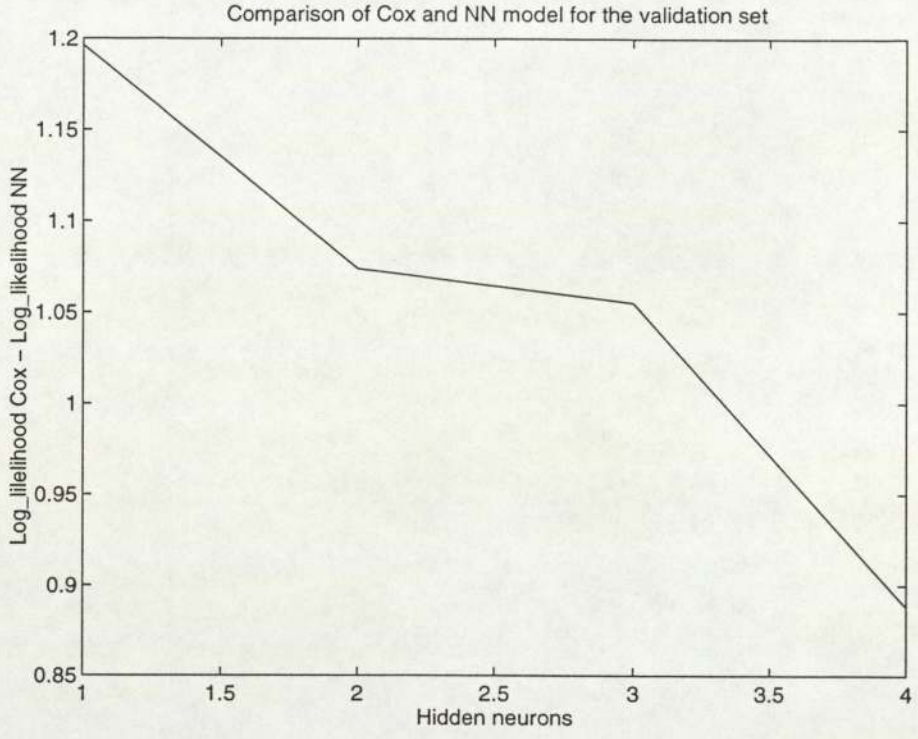


Figure 4.17: Comparison between Cox method and the neural network method for the validation set. It actually represents graphically the relation: $Log\_likelihoodCox - Log\_likelihoodNN$

The parameters of the network are described by a vector of dimension 3. The first element indicates the number of inputs, that is, the number of the prognostic factors used, the second element indicates the number of hidden neurons and the third, the number of outputs, which in our case is always 1.

As seen from table 4.2 the result of the linear case for the training set is always smaller than the ones obtained from the neural networks. In fact, from the different constructions of the networks, we see that the error value for the training set is rather similar for the cases of the 1, 2 and 3 hidden neurons. In the case of four hidden neurons, the result is quite smaller than the other cases, but is still larger. Observing the results from the test set, one notices that they are a bit smaller than the linear case, but the difference is quite small.

To see the changes of the log-likelihood values when one of the prognostic factors is not included in the data set, we derived table 4.3. By excluding from the data set one prognostic factor at a time, the error values were calculated using the linear method and the four constructions of the neural networks. Observing the linear case results

and comparing them with the values presented in table 4.2, one notices that the values for the training set are bigger than the ones in table 4.2. This observation stands for all the exclusions of the factors. In the cases of the neural networks we observe that the values for the training set are generally smaller than the ones found in table 4.2. Only in the case of the network having 4 hidden neurons the training values are bigger. The test set values in general do not have much difference from the case of table 4.2.

Parameters of the network	Without Treatment		Without WHO index		Wihtout Primary Tumour		Without Dexamethasone		Without Age	
	-L training	-L test	-L training	-L test	-L training	-L test	-L training	-L test	-L training	-L test
Linear	2186.08	118.54	2191.47	119.16	2188.73	118.58	2185.84	118.46	2187.95	117.75
4 1 1	2197.56	118.49	2198.92	117.73	2198.93	117..53	2192.85	117.98	2197.78	117.77
4 2 1	2199.58	117.77	2194.46	118.14	2199.39	118.69	2197.10	118.98	2198.17	117.01
4 3 1	2198.56	117.02	2199.45	118.22	2196.76	117.91	2186.40	118.64	2191.47	119.33
4 4 1	2199.22	117.65	2199.56	117.77	2199.39	117.85	2193.94	117.96	2179.94	118.92

Table 4.3: Log-likelihood results using 4 networks constructions, and using four prognostic factors, that is excluding one prognostic factor at a time

Chapter 5

Estimated cumulative hazard and survival probability

The goal of the Cox regression method is to estimate the hazard rate of the patients of the trial, and draw conclusions from there. In this chapter, results are presented from the estimation of the cumulative baseline hazard function, first by using synthetic data and then using the real cancer data. The objective in this chapter is to estimate the cumulative hazard function $H(t)$ using the method of Cox and Oakes (1984) [1] and also in Christensen's paper (1987) [5].

5.1 Overview of hazard function estimation

5.1.1 Estimation method by Cox and Oakes

The established way of presenting survival data is to estimate the survival curve. In the case of the data that we have, since all the survival times are complete, meaning that there is no missing data, the survival curve is estimated simply as the proportion of individuals to whom the event has not yet occurred at each point of time during the observation period (Cox & Oakes [1]).

To estimate the cumulative hazard rate $H(t)$, we first need to estimate the cumulative baseline hazard $H_0(t)$ which is given by

$$H_0(t) = \int_0^t h_0(u)du \tag{5.1}$$

As suggested by Cox and Oakes, we can compute $H_0(t)$ non parametrically, by noting that the total number of failures in the time interval (0,t) is

$$D(t) = \sum_{\tau_j < \tau} d_j \tag{5.2}$$

where d_j is the number of deaths at time t. This implies that we take into account the fact that there might be more than one death at time t.

So the cumulative baseline hazard is suggested to be estimated by

$$\hat{H}_0(t) = \sum_{\tau_j < t} \frac{d_j}{\sum_{\ell \in \Re(\tau_j)} \hat{\psi}(\ell)} \tag{5.3}$$

where $\hat{\psi}(\ell)$ is a vector including the estimated values of $\psi(\ell)$.

The estimated baseline hazard for time t_i, is the number of deaths that occurred at that specific time t, divided by the estimated hazard rate of the patients that belong to the risk set. Therefore, the cumulative baseline hazard is the sum of the baseline hazards up to time t.

The estimated hazard function can be found by using

$$\hat{H}_i(t) = \hat{\psi}(i)\hat{H}_0(t). \tag{5.4}$$

.

It can also be estimated by the total survivor function, or the cumulative survival probability denoted by $\hat{F}_i(t)$, which can be determined as:

$$\hat{F}_i(t) = [\hat{F}_0(t)]^{\hat{\psi}(i)} \tag{5.5}$$

where $\hat{F}_0(t)$ is the estimated baseline survivor function and is given by

$$\hat{F}_0(t) = \exp[-\hat{H}_0(t)]. \tag{5.6}$$

.

With the aid of these estimators, graphical illustrations of the estimated effects of the prognostic variables on survival time can be given.

This method of estimation as found in Cox and Oakes can be established by using both approaches, i.e., the Cox method and the neural network method.

5.2 Implementation of toy problem 3

In order to evaluate the performance of the approach, we tested the methods, using the standard Cox method and the Cox method with neural networks, on synthetic data.

5.2.1 Implementation using the Cox model

For this toy problem, the same synthetic data as that in toy problem 1 described in section 3.1 is used. The purpose of this toy problem is to determine the success of the implementation by re-evaluating the initial estimated cumulative baseline hazard function.

The procedure followed for the implementation of this toy problem is the following:

- The synthetic data was generated exactly in the same way as in toy problem 1 and so it is a normal distribution.
- β vector is chosen to be a vector β_{init} that again is normally distributed.
- The survival times were estimated and sorted in ascending order, using the method described in toy problem 1, and the data was sorted according to them.
- Having the data, the estimated cumulative baseline hazard was computed and it was called $H_0(t)_{init}$.
- The β vector was again initialised to some random vector.
- By using the scaled conjugate gradient the β vector is optimised, and β_{fin} is obtained.
- The estimated cumulative baseline hazard is then calculated and is called $H_0(t)_{fin}$.
- The initial and the final estimated cumulative baseline hazard are compared and if they are similar then it means that the implementation used is successful.

Figures 5.1 to 5.4 show the results when using 10, 100, 1000 and 2000 input points.

As observed from the figures 5.1 to 5.4 as the number of points increases, the approximation to the initial $H_0(t)$ gets closer and closer. In figure 5.4, which is the one

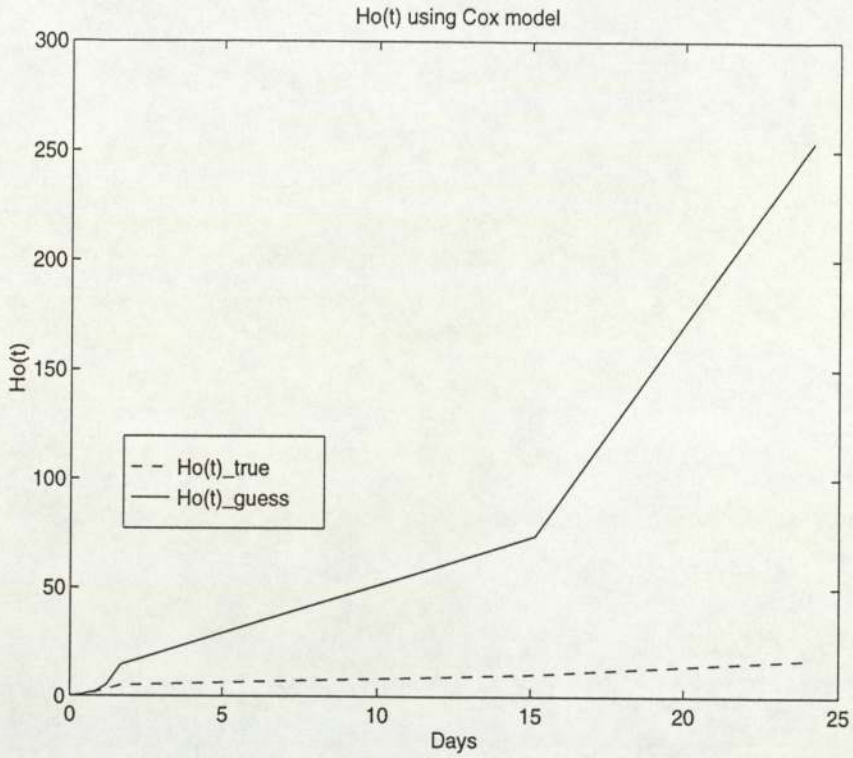


Figure 5.1: Estimation of $H_0(t)$ on a synthetic data with Cox model, using 10 input points

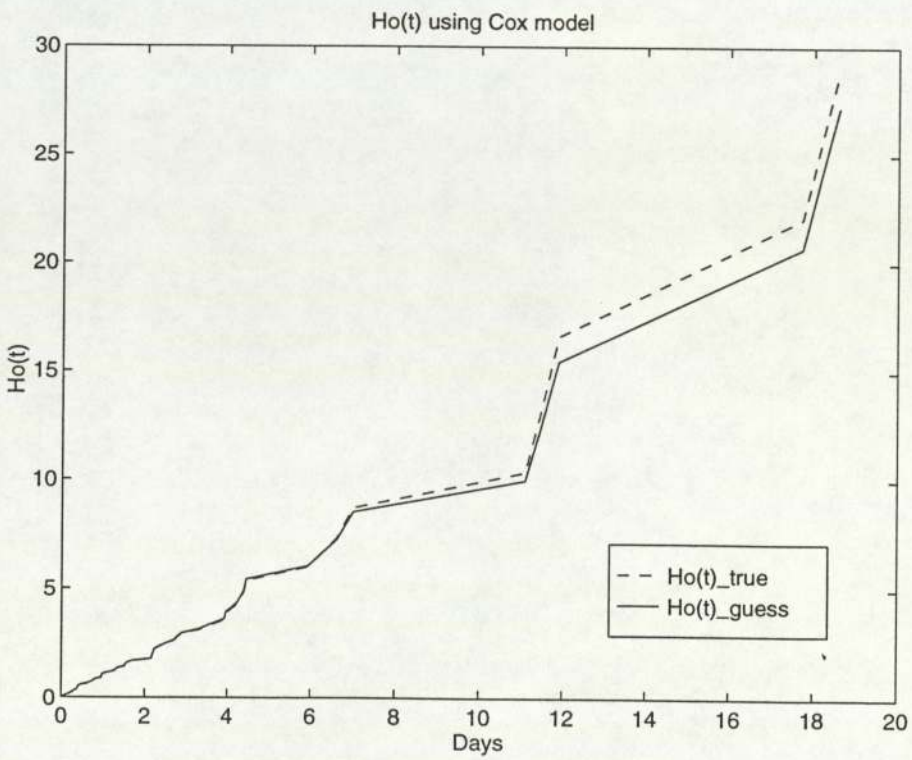


Figure 5.2: Estimation of $H_0(t)$ on a synthetic data with Cox model, using 100 input points

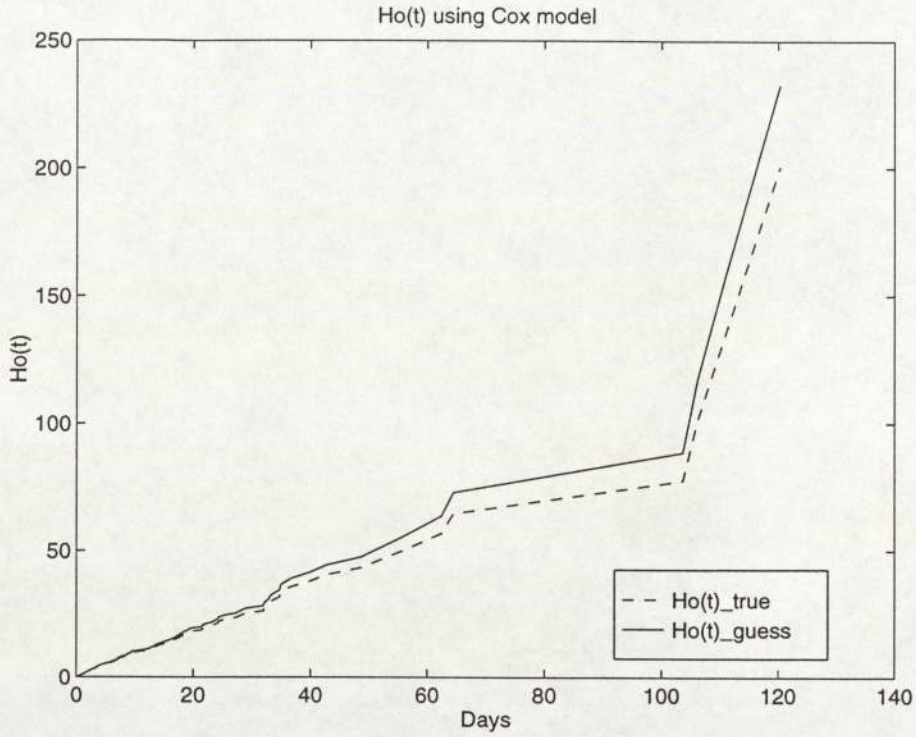


Figure 5.3: Estimation of $H_0(t)$ on a synthetic data with Cox model, using 1000 input points

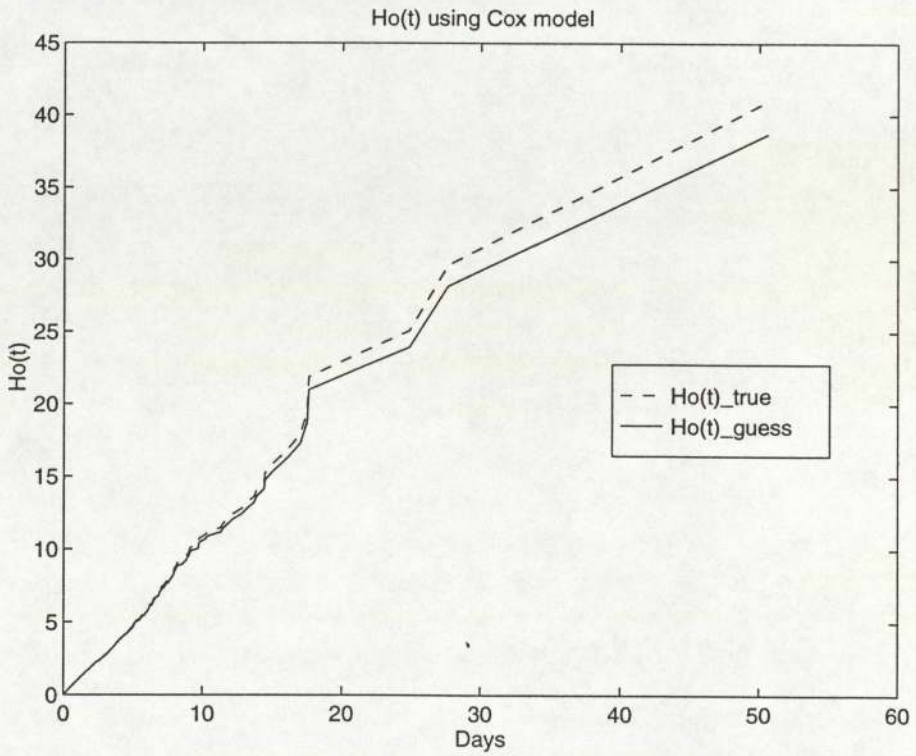


Figure 5.4: Estimation of $H_0(t)$ on a synthetic data with Cox model, using 2000 input points

where 2000 input points are used, the best approximation is obtained from among all the cases. In the other cases though, of the 100 and 1000 input points, the approximation was not so bad, in fact it was fairly accurate. The worst case is found of course in figure 5.1 which uses 10 input points. This shows that the method applied here using the standard Cox method, seems to be correct.

5.2.2 Implementation using neural network model

This implementation of the toy problem, using the neural network method is actually a continuation of toy problem 2 which is described in section 4.2. The way that the data was generated is the same as in toy problem 2.

The procedure followed for this part of the toy problem is the following:

- The synthetic data is generated exactly the same way as in toy problem 2. So it has one dimension, length N, and is normally distributed.
- y_{init} is chosen in this case to be a simple function $f(x)$, such as $\sin x$.
- Knowing y_{init} the ψ vector is then computed, and so the $H_0(t)_{init}$ is calculated.
- The parameters of the network are initialised by constructing a two layer feed-forward neural network.
- Then the weights are optimised using scaled conjugate gradient.
- The output of the network y_{fin} is then derived.
- ψ vector is calculated and then $H_0(t)_{fin}$ is derived.
- A comparison of the two baseline functions will show if the implementation is successful.

As expected, this toy problem will prove as well that by increasing the number of input points and hidden neurons up to a point, the networks performance improves. Keeping in mind the results of toy problem 2, the implementation here was tested for 10, 1000 and 2000 input points, as the programs require a lot of execution time. The results are presented in figures 5.5 to 5.7.

In figure 5.5 one observes that for 100 points and 3 hidden neurons, the networks guess is quite far from $H_0(t)_{init}$. In figure 5.6 were the number of input points is 1000 and the network has 3 hidden neurons, the prediction of the true $H_0(t)$ is much better. The offset is smaller than the previous case but still is quite large. The next experiment is presented in figure 5.7 and the result is a bit better than in figure 5.6. Here, 2000 input points and 3 hidden neurons are used. One notices that the performance of the network is better. The offset between the two curves is now quite small, and the shape of the predicted graph is now close to the initial one. Of course, as expected, the performance of the network gets better as the number of input points are increased. In the experiments of this implementation the number of hidden neurons were kept the same, although from the previous toy problems it is noticed that as the number of hidden neurons increases the network has better performance.

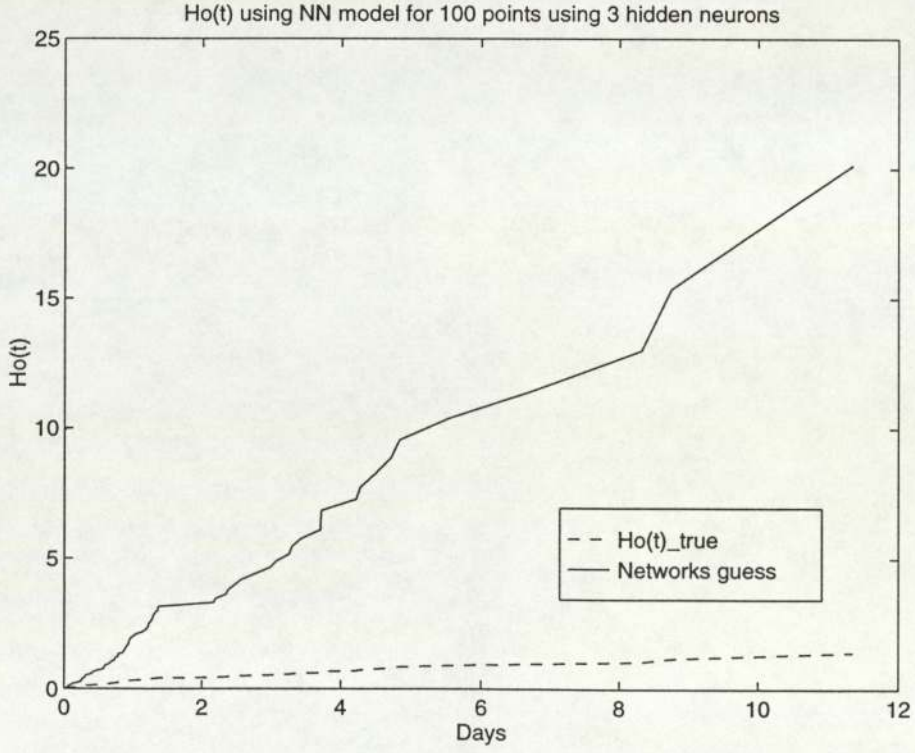


Figure 5.5: Approximation of $H_0(t)$ using the NN model with 100 inputs and 3 hidden neurons

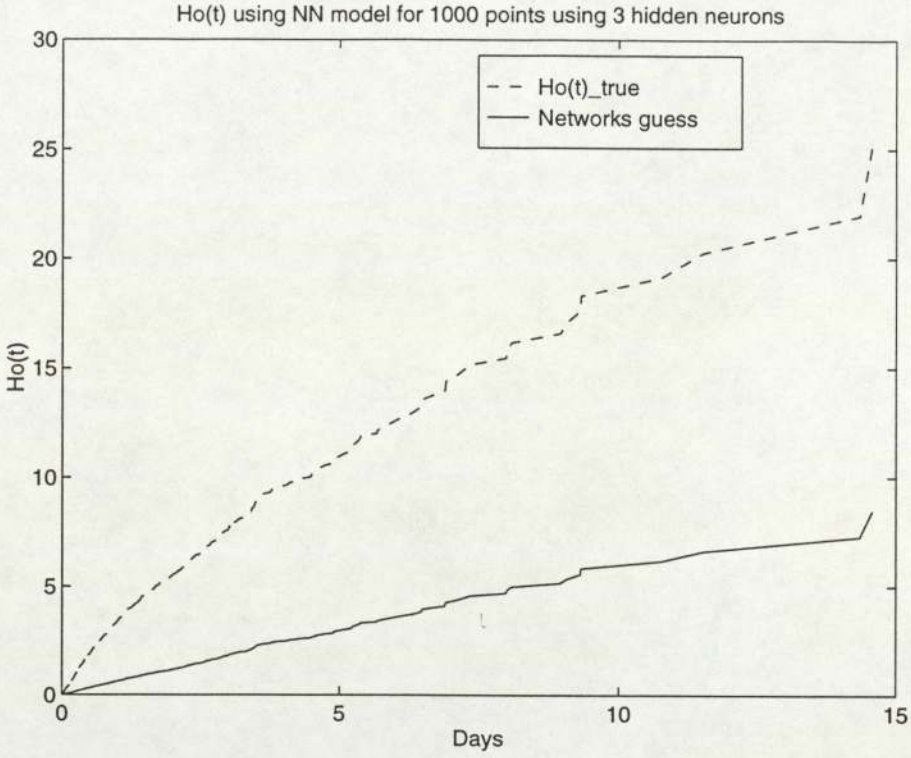


Figure 5.6: Approximation of Ho(t) using the NN model with 1000 inputs and 3 hidden neurons

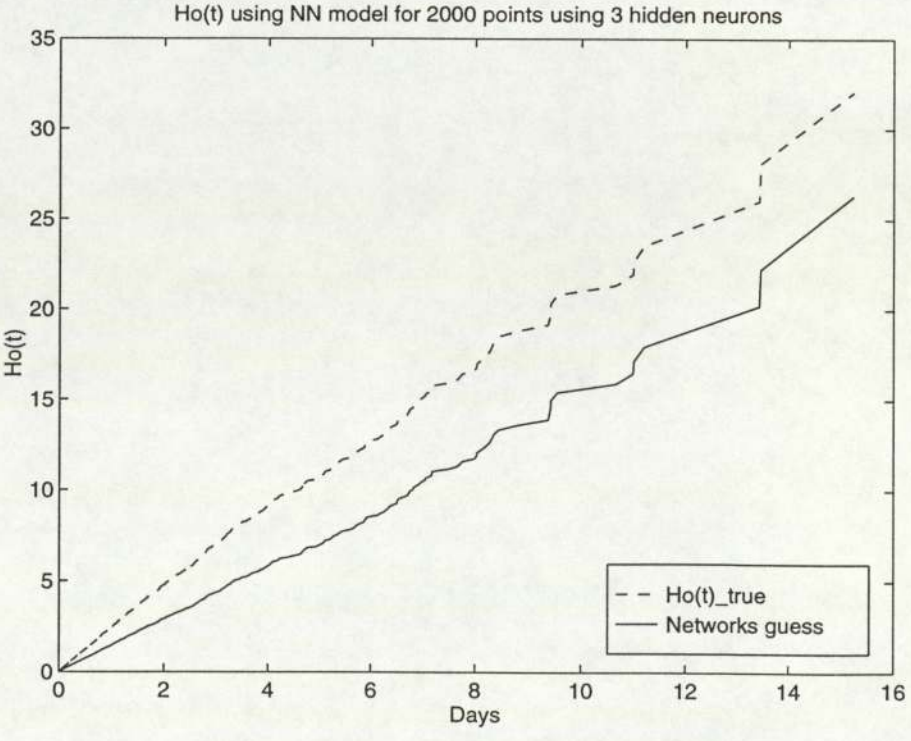


Figure 5.7: Approximation of Ho(t) using the NN model with 2000 inputs and 3 hidden neurons

5.3 Implementation on the real data

In this section, we present the results of the implementation of both methods, Cox and neural networks, on the real data, in order to estimate the cumulative hazard function of the patients.

As observed from the figure as the number of inputs increases, then the estimation of the network is improves. This shows that the implementation to derive the cumulative baseline hazard seems to be successful.

5.3.1 Implementation using the Cox model

By using equation 5.3, the cumulative baseline hazard $H_0(t)$ is derived from the real data. In this implementation, the data set used is one that has the actual values of the prognostic factors. It consists all the 5 prognostic factors mentioned in section 3.2.

Figure 5.9 shows the cumulative baseline hazard found from this implementation.

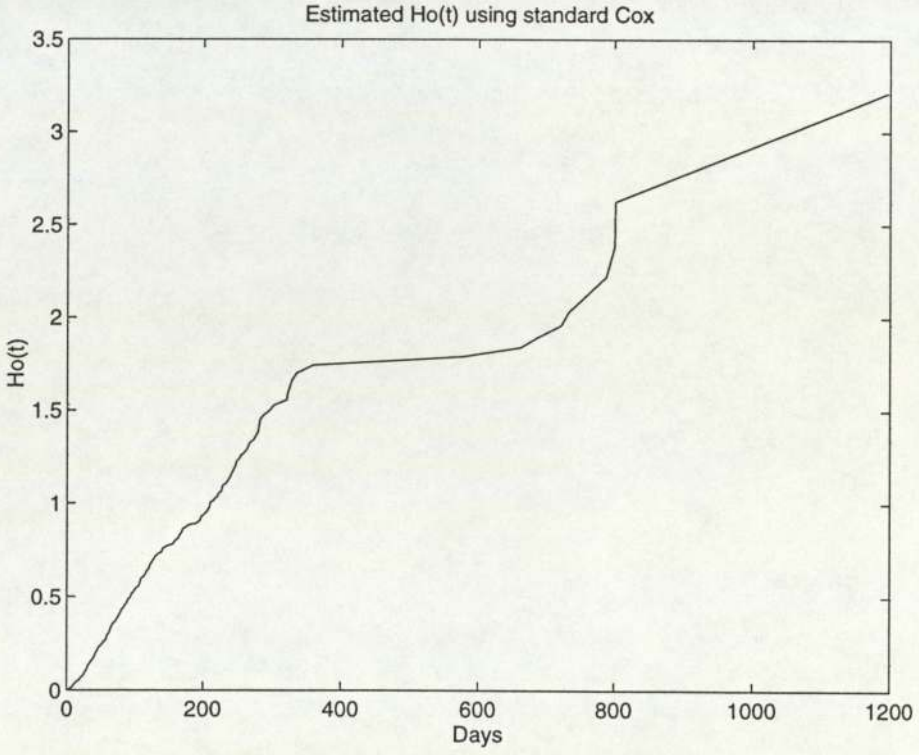


Figure 5.8: Estimation of $H_0(t)$ of the real data with Cox model

As figure 5.9 shows, after about 350 days of survival time, the cumulative baseline hazard stabilizes, then it increases smoothly and after about 900 days it increases

steeply. This happens because after 350 survival days there are not many patients that are still alive and that is confirmed by figure A.8 and by the histogram in figure 1.1. So the first part of the graph, until the 350 days, increases quite steeply and is has a ramh form, which is expected.

Knowing $H_0(t)$ and by applying equations 5.4, 5.5 and 5.6, it is quite easy to estimate the cumulative hazard function $H(t)$, and the survival probability $F(t)$. These estimations are presented in figure 5.10.

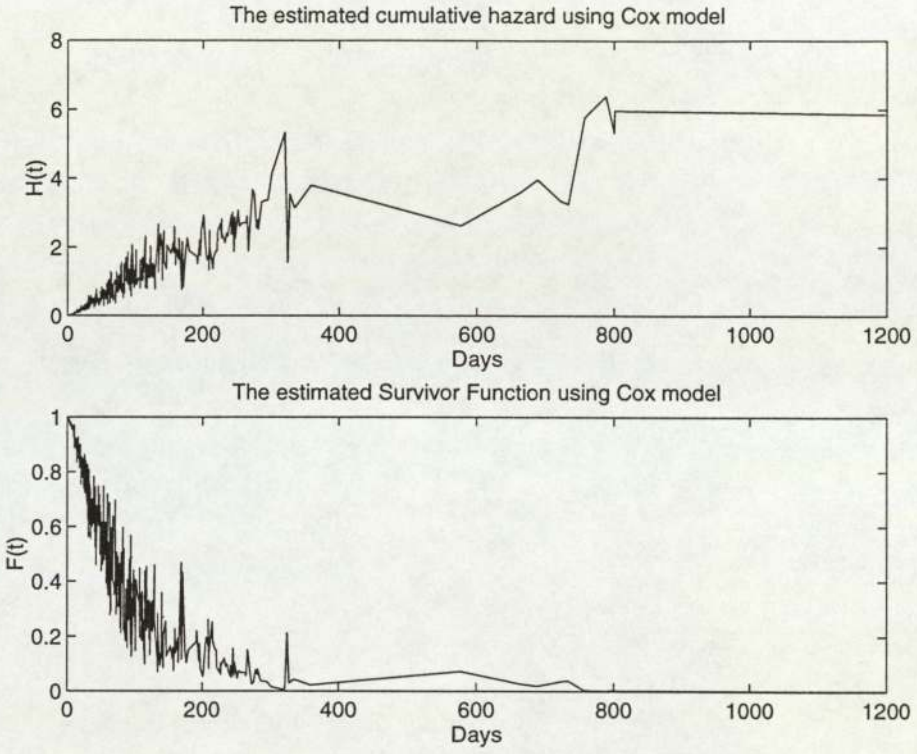


Figure 5.9: Estimation of $H(t)$ and $F(t)$ of the real data with Cox model

Observing figure 5.11 shows that the estimated values for $H(t)$ and $F(t)$, are as expected. The bottom graph that presents the survival probability $F(t)$ is a ramh curve which drops off quite steeply until the 350 days, as it should. After that point, it becomes closer to zero, since very few patients are alive at the interval after 350 days. Also $H(t)$ increases steeply in a fluctuating form until 350 days, and then gets a bit smoother. That is expected again, because an increasing hazard results in a decreasing survival probability.

5.3.2 Implementation using Cox method with neural networks

The equations used in this implementation are the same as the ones used with the standard Cox model. What differs is that now we use the generalised Cox formalism for ψ as given by equation 4.2. It was tried with different network constructions, using 1, 2, 3 and 4 hidden neurons.

Figure 5.10 presents the cumulative baseline hazard $H_0(t)$ when using a network with 1 hidden neuron. The result obtained is very close to the $H_0(t)$ found when using the standard Cox method. The values of the two estimations are not very different, as observed from the comparison of the figures 5.8 and 5.10.

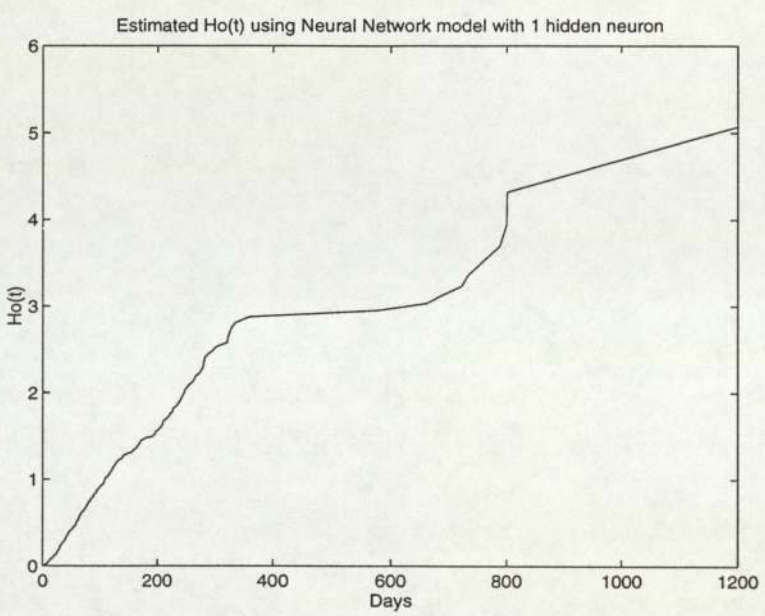


Figure 5.10: Estimation of $H_0(t)$ of the real data with Cox model using neural networks with 1 hidden neuron

Figure 5.11 represents the estimated cumulative hazard and the estimated survival probability. Both graphs are much smoother compared to the ones obtained when using the Cox model in figure 5.9.

In figures 5.12 to 5.17 we present the results obtained when having 2, 3 and 4 hidden neurons in the network.

Judging from figures 5.12 to 5.17, there is not much difference in the values obtained for the estimated baseline hazard, compared to the Cox model results. Generally, one notices that the curves for the estimated cumulative hazard and survival probability are smooth, unlike the ones obtained using the standard Cox model which have a

ASTON UNIVERSITY
LIBRARY & INFORMATION SERVICES

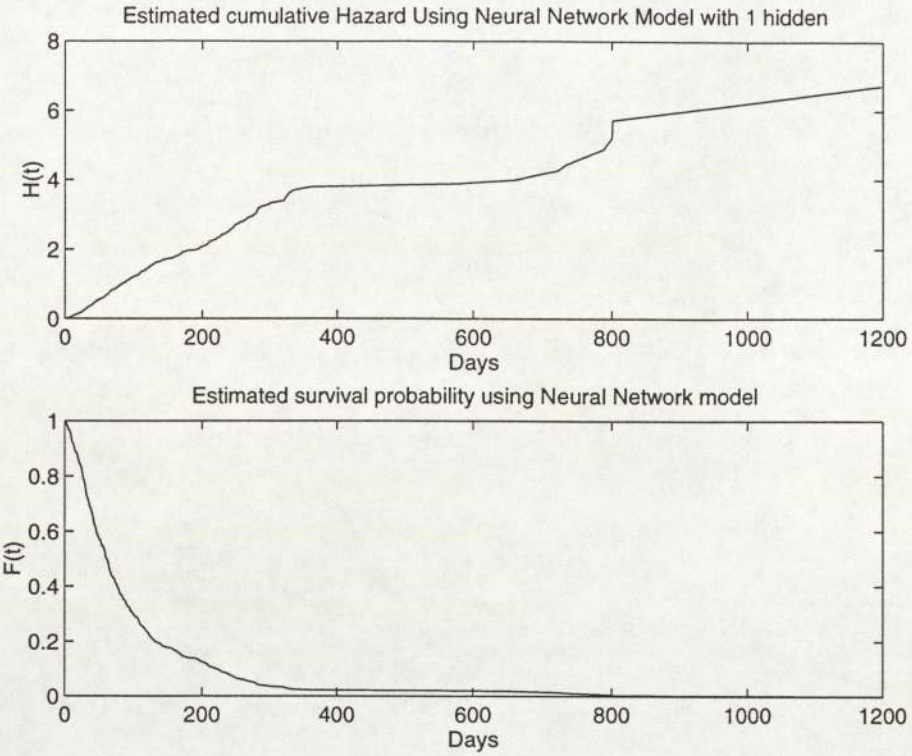


Figure 5.11: Estimation of $H(t)$ and $F(t)$ of the real data with Cox model using neural networks with 1 hidden neuron

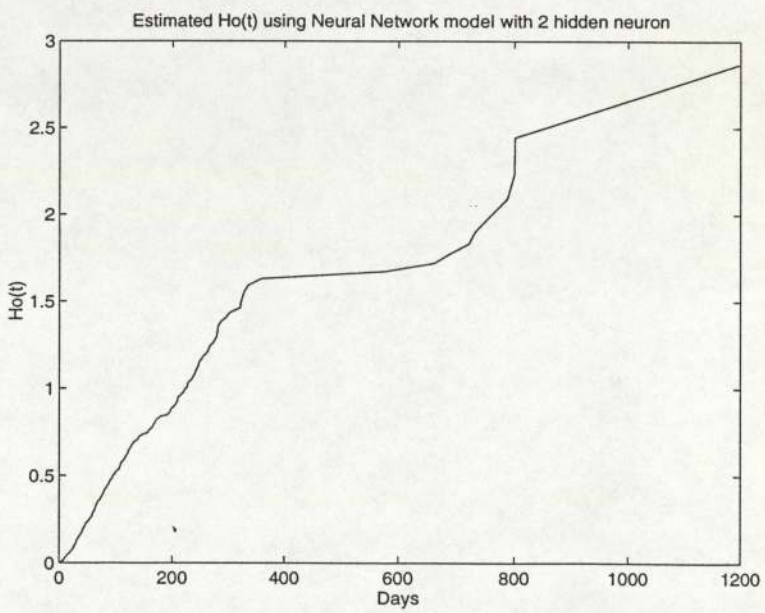


Figure 5.12: Estimation of $H_0(t)$ of the real data with Cox model using neural networks with 2 hidden neurons

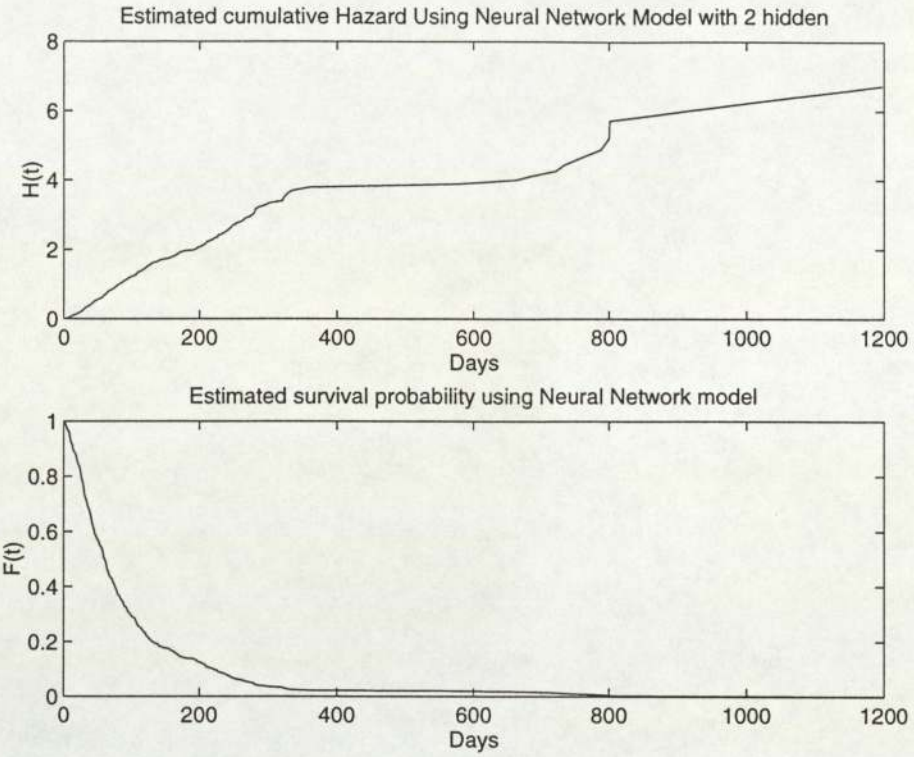


Figure 5.13: Estimation of $H(t)$ and $F(t)$ of the real data with Cox model using neural networks with 2 hidden neurons

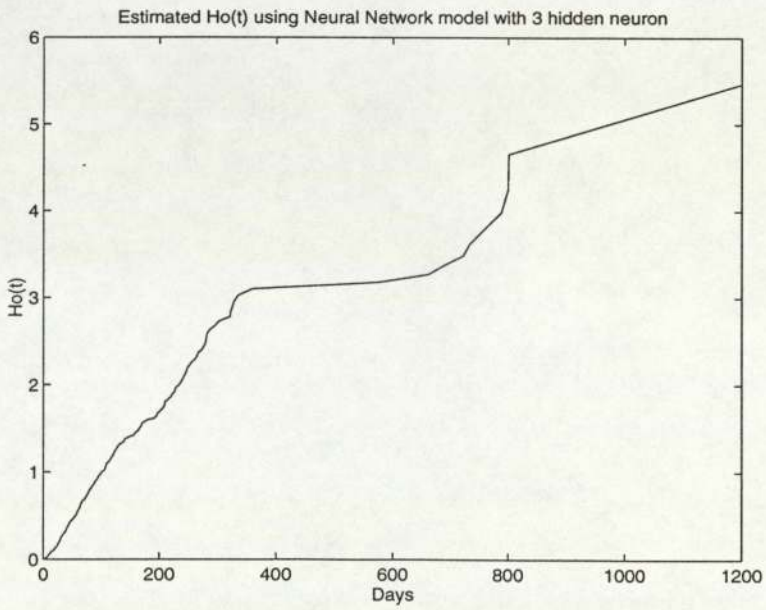


Figure 5.14: Estimation of $H_0(t)$ of the real data with Cox model using neural networks with 3 hidden neurons

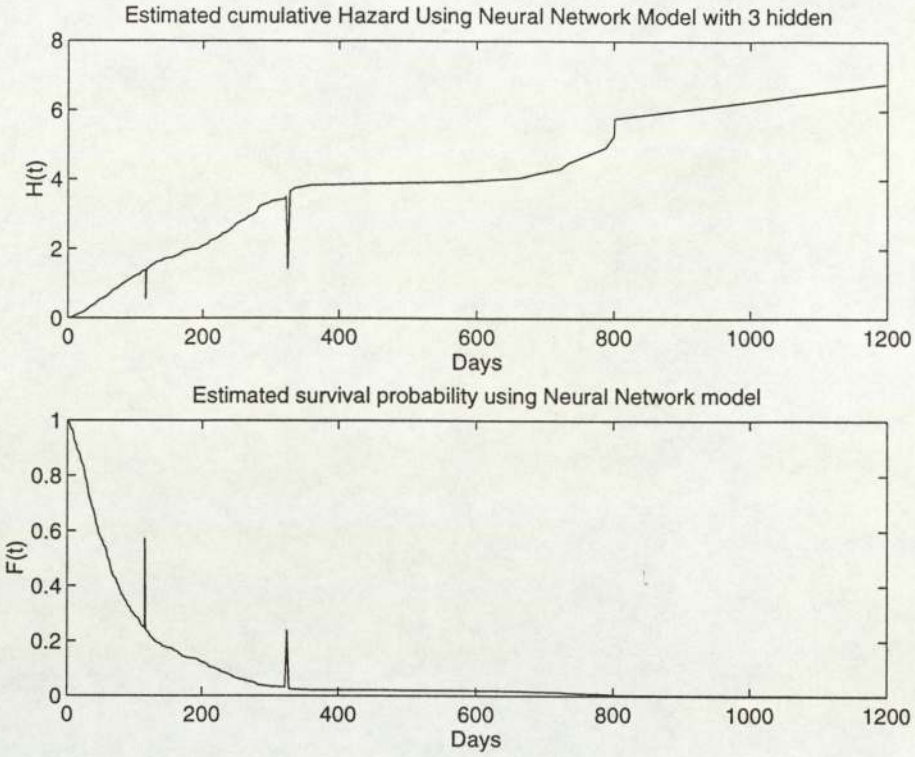


Figure 5.15: Estimation of $H(t)$ and $F(t)$ of the real data with Cox model using neural networks with 3 hidden neurons

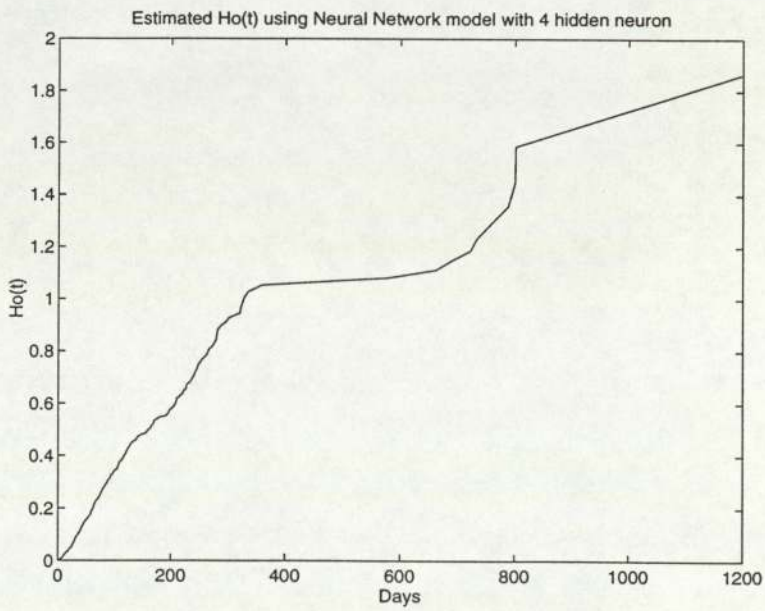


Figure 5.16: Estimation of $H_0(t)$ of the real data with Cox model using neural networks with 4 hidden neurons

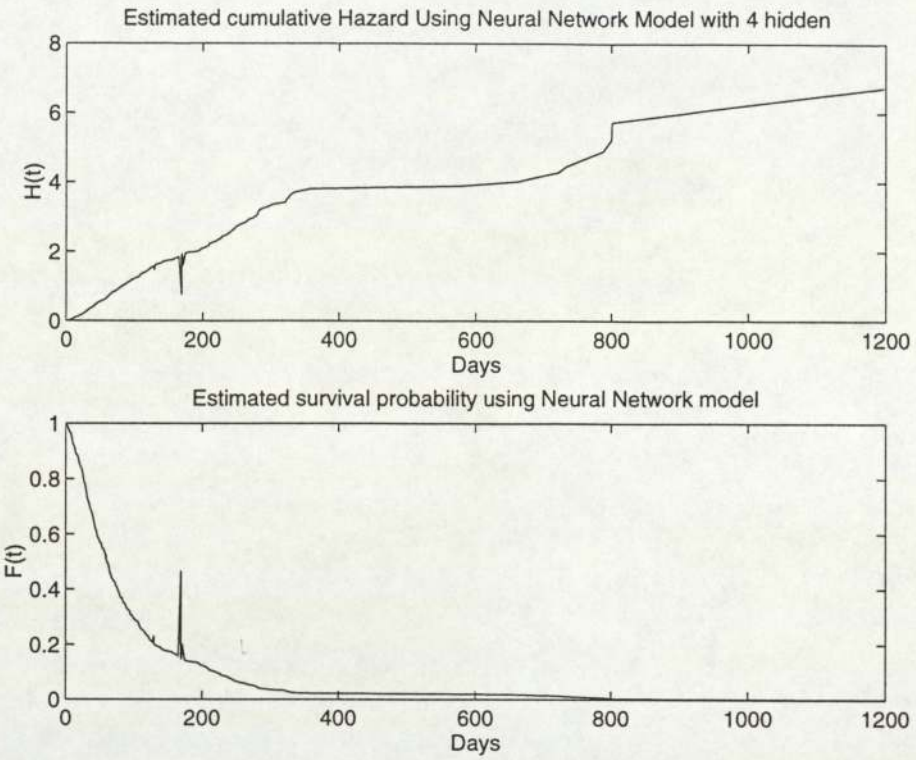


Figure 5.17: Estimation of $H(t)$ and $F(t)$ of the real data with Cox model using neural networks with 4 hidden neurons

fluctuating ramh form.

These results indicate that the Cox method using neural networks is similar in performance to the standard Cox method, in estimating the cumulative baseline hazard, and also the cumulative hazard and survival probability. That means that, in this case, both methods can be used for the estimation of the cumulative hazard and survival probability. The encouraging result here is that we know that the neural network approach, which is the new method introduced, is capable of at least estimating similar results as the standard Cox regression method for this particular application.

Chapter 6

Discussion

It is already known that the Cox regression model is a powerful statistical tool for survival data analysis. It is the principal method used up to now by statisticians in clinical trials.

In this project we attempted to introduce a new approach, based on the Cox regression model that includes neural networks. As one can conclude from the several implementations presented in the project, the results obtained with the new approach are comparable with the ones found by using the standard Cox regression method. All the implementations of the new approach were first tested on synthetic data, so as to gain confidence that the implementation is correct.

For that reason, toy problem 2 and toy problem 3 were created. As mentioned in section 4.2, in toy problem 2 the network tries to compute the hazard function of the patients, given their prognostic factors and survival times. So by having the correct survival function, when a new patient comes along, then by knowing his/her prognostic variables, we would be able to predict his hazard and in effect his/her estimated survival.

In chapter 5, it was attempted to estimate the cumulative hazard function and cumulative survival probability of the patients. It is interesting to use equation 5.1 to find the baseline hazard $h_0(t)$, so that it can be used in estimating the hazard and survival probability for the individual patients, knowing his/her prognostic variables. Such an implementation, however, is computationally very time consuming. Usually, the statisticians use a standard package for survival data analysis,which computes the

baseline hazard $h_0(t)$. By studing the user manual for such packages, one can find the way that is computed in the package and apply it using the Cox method with neural networks. Because of a lack of time, this estimation was not attempted in this project.

Concerning Cox regression, the likelihood expression for the survival function varies according to the hypothesis. The hypothesis used for these implementations, as mentioned in section 2.2, is that the likelihood function is the probability that all the patients in the set die. The same hypothesis was used to derive the likelihood expression for the neural networks case. A test was designed to compare the likelihood values obtained when using the linear and the neural network model in the Cox regression. A test was also designed to compare the likelihood values, when one prognostic variable was taken out from the set one at a time, by using the two approaches. After performing the cross-validation procedure the data set was split into a training and a test set. The log-likelihood values of the two sets were calculated using both approaches and they were compared. For the neural case, network four network constructions were tested, having 1, 2, 3 and 4 hidden neurons. The error values obtained for the training set of all the cases of the neural networks were greater than the value using the linear case. This result is quite encouraging, as it means that by using the new approach, namely, standard Cox with neural networks, the specific data set can be modelled better than in the linear case. Also, in the case in the test set, the difference in the error values for the two approaches is very small; actually they are almost similar. In general, it can be said that whether neural networks can improve the goodness of fit or not is still undecided for this example and in general.

The big problem, however, that we concluded from the implementations of the new approach in this project is the one derived from the toy problems. It is the fact that in order for the neural network approach to achieve accurate results, it needs a large number of input points. Such an amount of input data is of course not possible to be found in clinical trials, as the clinicians do not deal with so many patients in the trials. So it can be said that for this particular application where if all the patients of this data set are dead, the new method is not going to be as successful. As mentioned before, it is good if the method is tested on new data, and performance is compared.

The problem that is faced by the statisticians in clinical trials is that the small

amount of data in cerebral metastases makes prior knowledge very important. That means that the statisticians must be able to select the right prognostic variables that are going to be used for the analysis. If too many prognostic factors are used, then neither a neural network nor a Cox survival function will be able to produce a good estimation of the real survival function.

The conclusions and suggestions that comes from this research are that to be able to understand the advantage of the new approach, it should be applied on several data sets, and the results compared with the standard Cox method. As mentioned before the new approach, namely, Cox regression using neural netwroks is not suitable for this specific application. When it is applied, however, on a bigger and different data set, then there are might be satisfactory results. If neural networks prove to be efficient in modelling survival functions of different survival data sets, then it will be a powerful tool for the statisticians to use. It will compensate for the lack of prior knowledge and the results will be more accurate.

Appendix A

Survival curves

To verify the results of the log-rank test, it is better to examine the true survival curves of each prognostic factor. In this way, we will have a clearer graphical way to determine the best category of each prognostic factor.

Figures A.1 to A.7 present the survival taking into account each prognostic factor.

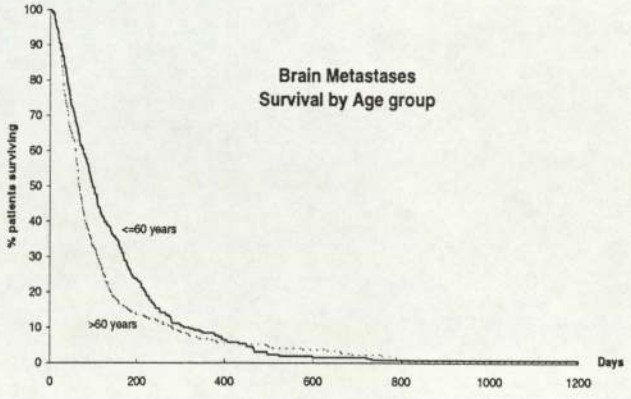


Figure A.1: Survival by Age of the patients

As seen in figure A.1 those patients that are less than 60 years old survive more than those that are older than 60 years. So based on this graph and on the exp/obs value of the first category, given in table 1.2, it seems clearly that the first category is the best.

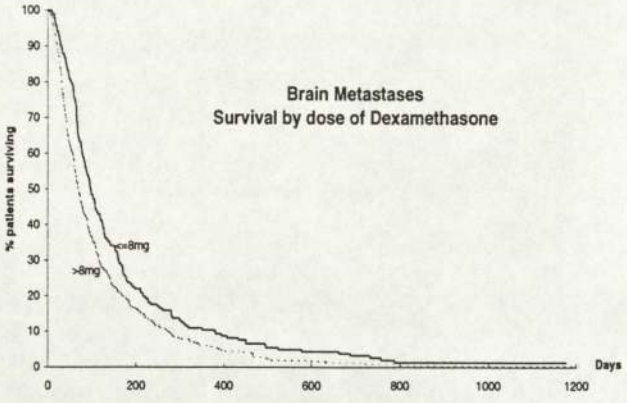


Figure A.2: Survival by the dexamethasone dose taken by the patients

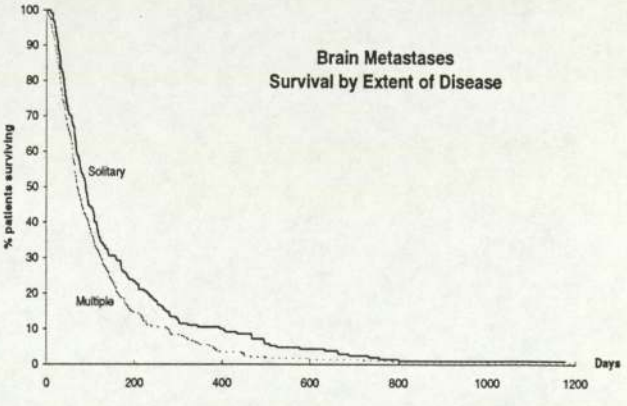


Figure A.3: Survival by the patients extend of the disease

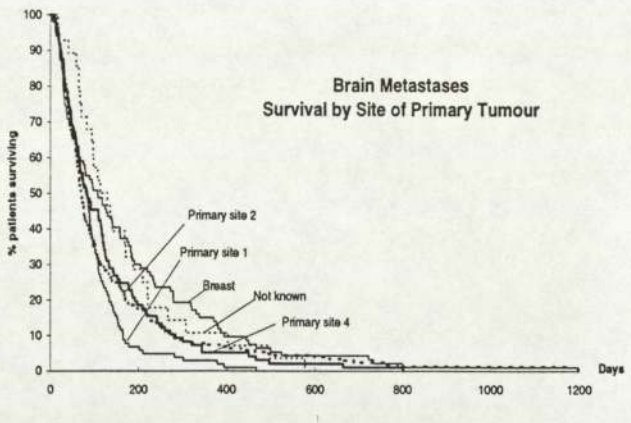


Figure A.4: Survival by the patients primary tumour

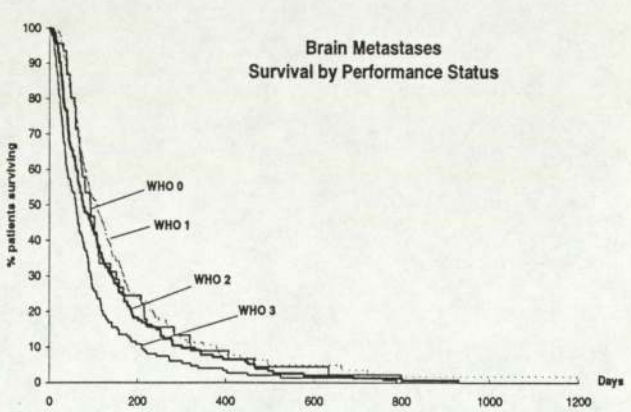


Figure A.5: Survival by the WHO index of the patients

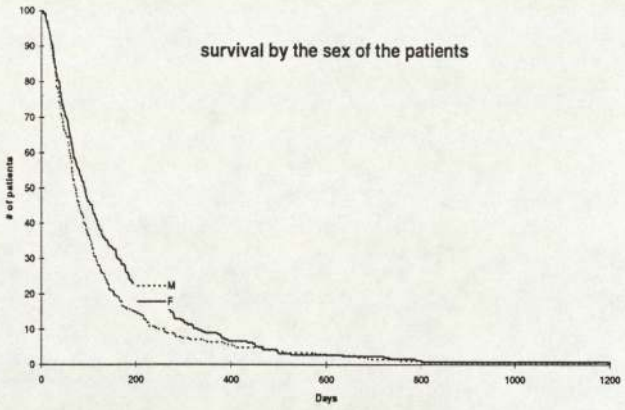


Figure A.6: Survival according to the patients sex

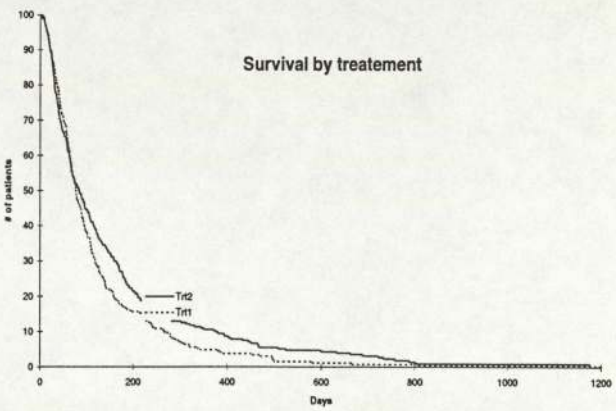


Figure A.7: Survival by the treatment each patient is receiving

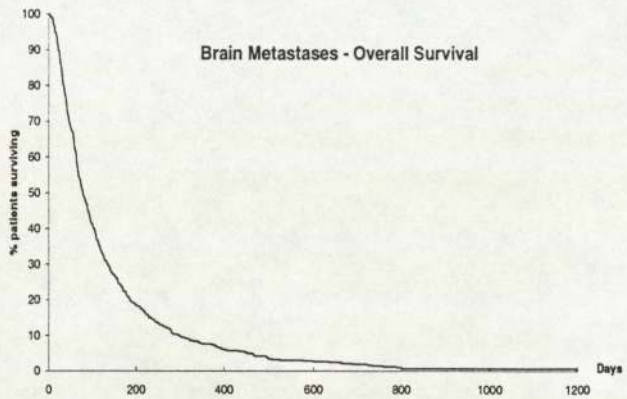


Figure A.8: The overall survival of the patients

The same stand, when observing figures A.2 to A.7. All the categories for the prognostic factors that were found to be the best according to the log-rank test as presented in table 1.2, now we see that the survival by those categories is better then the others of the prognostic factor.

Bibliography

[1] Cox D. R. and Oaks D. *Analysis of survival data.* Chapman and Hall, London 1984.

[2] Mashesh K. B. Parmar and David Marchin, *Survival analysis: a practical approach.* Willey, Chichester, 1995, pg. 121 - 142.

[3] Christopher M. Bishop *Neural networks for pattern recognition*, Clarendon press, Oxford, 1995.

[4] Elisa T. Lee, *Statistical methods for survival data analysis*, Wiley, New York, 1992, pg. 109-113.

[5] Erik Christensen, *Multivariate survival analysis using Cox's regression model*, Special articles, 1987.

[6] T J Priestman, *The Royal College of Radiologists - different dose/fraction regimes in cerebral metastases*, Birmingham 1989.

[7] Christopher M. Bishop, Francois Collet, Janet Dunn and Christopher Poole, *Neural networks for cancer prognosis* article, 1996.

[8] L. Tarassenko, R. Whitehouse, G. Gasparini and A. L. Harris, *Neural network prediction of relapse in breast cancer patients*, Neural Computing and Applications, London, 1996.

[9] Douglas G. Altman, *Practical statistics for medical research*, Chapman and Hall, London, 1992.

[10] B. Borgelt, R. Gelber, M. Larson, F. Hendrickson, T. Griffin and R. Roth, *Ultra-rapid high dose radiation for brain metastases*, Radiation Oncology, Volume 7, Number 12, 1981.

[11] J. West and M. Maor, *Intracranial metastases*, Radiation Oncology, Volume 6, Number 1, 1980.

[12] Lawrence R. Coia, *The role of radiation therapy in treatment of brain metastases*, Radiation Oncology, Volume 23, Number 1, 1992.

[13] M. D. West, T. Dobbins, T. Phillips and D. Nelson, *Brain metastases optimal subgroup*, Radiation Oncology, Volume 16, Number 3, 1989.

[14] *Statistical methods in medical research*,Blackwell scientific publications, London, 1971.

[15] A. J. Gross and V. A. Clark, *Survival distributions: reliability applications in the biomedical sciences*, Wiley distributions, New York, 1975.

[16] BMDP manual.

[17] B. Borgelt, R. Gelber, s. Kramer, L. Brady, C. Chang, L. Davis, C. Perez and F. Hendrickson, *The palliation of brain metastases: final results of the first two studies by the radiation therapy oncology group*, Radiaton Oncology, Volume 6, Number 1, 1980.