A Study of Patient-specific Prognosis of Ovarian Cancer

Nicolas Charles

MSc by Research in Pattern Analysis and Neural Networks

ASTON UNIVERSITY

September 2002

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.

Acknowledgements

I would like to thank particularly my supervisors, Professor David Lowe and Doctor Jort van Mourik, for their help and especially Professor D. Lowe for his patience, and without whom this work would not have been possible. Many thanks also to Dr Judy Powell and Dr Sean Kehoe of the Birmingham Women's Hospital for the data they provided us.

I am grateful to all my lecturers this year for the quality of their lectures, Prof. D. Lowe, Prof. D. Saad, Dr I. Nabney, Dr M. Opper, Dr I. Stainvas and Dr P. Tino.

I would also like to thank a lot all the MSc students in PANN, Remi, Manu, Boremi, Oliver and Mani for their support and friendship, as well as the members of the NCRG, Stephane, Lehel, Dan, David, Wei Lee, Vicky and Christopher.

I would also like to thank my friends for their support, Alexandre, Amine, Amos, August, Chris, Clement, Delphine, Eric, Jenny, Jimmy, Katherine, Liz, Sebastien, Sergei, Shumsa and Sophie who were there this year at Aston, and also Bruno, Gilles, Laure, Matthias, Raphael, Vitali and Vivek for their visit.

And finally, I thank my parents for their love and financial support which made this year possible.

ASTON UNIVERSITY

A Study of Patient-specific Prognosis of Ovarian Cancer

NICOLAS CHARLES

MSc by Research in Pattern Analysis and Neural Networks, 2002

Thesis Summary

Current cancer prognosis are based on broad population averages statements. This thesis, focused on ovarian cancer, aims to estimate patients survival time. Different Neural Networks are tested on a medical dataset containing physiological information on patients. First predictions on the survival time are obtained by standard point estimators such as Multilayer Perceptrons (MLP) and Radial Basis Function (RBF) networks. But as the results are quite disappointing, a novel estimation technique is introduced: Mixture Density Networks (MDN). The MDN method provides a probabilistic model for the estimation which cannot be obtained by others methods. Hence we obtained the full distribution of the probabilities of the survival time and discovered that it is highly multimodal, so no reliable prediction can be made. Indeed, the error rate obtained with the best model is about 70 %. Finally, some attempts at classifying patients into different classes of survival time are made, and the results are quite surprising as the Neural Networks can only distinguish censored patient and patients with deadly outcome.

Keywords: Medical study, Cancer Prognosis, Mixture Density Network

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Cancer is one of the most deadly of diseases. According to Cancer Research UK, one in three people will be diagnosed with a cancer during his/her lifetime. Each year, nearly 7000 new cases of ovarian cancer are diagnosed. When treated early, the survival rate for this specific cancer is quite good with nearly 80 percent of 5-year survival rate according to Ovacome[1], but unfortunately the survival rate decreases dramatically with the advance of the cancer. Indeed patients diagnosed with very advanced ovarian cancer have a less than 15 percent chance to survive for five year.

Currently, feedback about survival time of patients is estimated using the expert knowledge of the medics, who can only deliver broad-group averages statements rather than patient-specific ones. Being able to estimate the survival time of a patient with limited medical data would greatly help the doctors in prescribing the appropriate treatment balancing quality of life, risks and side effects from various treatments for patients suffering from ovarian cancer.

The aim of the research in this thesis is to see to what extent it is possible to estimate the likely survival time for ovarian cancer sufferers. Given the fact that current medical practice is unable to estimate patient-specific prognosis for this ailment, the aim of this thesis is speculative and ambitious, but if partially successful, could have significant benefit.

The Birmingham Women's Hospital provided us with a large (by medical standards) dataset of 1426 patients with 35 variables. But as with most medical datasets, it is subject to problems of noisy and missing data, and pointing out the most relevant

[1] www.ovacome.org.uk

variables is not easy.

The issue of reliability should be raised, to ensure that the prediction is actually suitable and will really help both medics and patients. Indeed, every prediction methods have to be tested and approved before being used in medical environment. But unfortunately, as the results of this thesis are not good enough, we did not explore any of risks estimation methods.

1.2 The dataset

The dataset used as part of this study, containing 1426 patient examples, with each patient record corresponding to a collection of 35 different variables, is the result of extensive data extraction from patient records taken over a 7-year period (1985-1992) in the United Kingdom. But as the data extraction finished, there were no follow-ups on people who survived the end of the study, and hence we do not know whether they survived the cancer, or for how long. In the following sections, this group of patients will be called *censored* patients. The variables of the dataset cover a very comprehensive questionnaire on the patients, ranging from demographic information, to medical and personal information. More information on the structure of this dataset is given in Appendix A.

Several different kinds of methods can be used on this dataset in order to try to estimate the survival time of the patients, but so far linear methods as well as human prediction provide poor results.

Indeed the survival time is obviously not an easy function of all the variables. Moreover the data is subject to noise. First of all there is noise on the inputs, as some variables are rather subjective. For example the stage is known to be sometimes over or underestimated[3]. Also the target value, the survival time, is subject to noise as well since we only know the minimum survival time of each patient. On the inputs, the noise will be assumed to be Gaussian, which is the standard assumption on noise models, but for the survival time we will see that it might not be the best assumption and a right-sided noise function will be used.

For all these reasons, the main estimation methods in this thesis for the survival time will be Neural Networks models, standard ones or combined with Mixture Models to obtain probabilistic models. Previous work on this dataset using Neural Networks has been conducted by Bruno Vincent in 1999[10], which demonstrated that it is a difficult prediction to make as the results obtained were quite disappointing.

In this thesis, we extend the range of techniques previously used, introducing some novel techniques to this domain which should also prove useful in other medical applications.

1.3 Thesis outline

The first part of this thesis discusses the preprocessing of the data, and mainly the missing data imputation using a noisy Independent Component Analysis. Then the question of the variables selection to lower the dimensionality of the problem will be addressed using non-linear methods and finally the different type of Neural Networks to be used to predict the survival time of the patients in both regression and classification approach will be discussed. In the regression case, we will especially study the Mixture Density Networks (MDN) which combines a Multilayer Perceptron with a Mixture Model to provide regions of conditional probabilities rather than the standard output of the MLPs and RBF networks which are just single point estimators rather than distributions. Moreover, this model allows the use of different noise models through different Mixture Models and hence we will explore the possibilities of using non-Gaussian or right-sided noise models. Finally, in the classification approach, MLPs will be used to try to separate the patients into different classes.

Chapter 2

Preprocessing of the data

This dataset consists of raw data extractions from patient records, so it contains dates, strings, codes, as well as continuous and semi-continuous values (the full details of the variables can be found in Appendix A). Hence it needs to be preprocessed to be suitable for automated machine learning techniques. Moreover, most of the patient records have at least some fields simply absent as the data was not collected, so all the missing values must be imputed for the same reason.

2.1 Variable modification

Some of the variables are continuous (for example, AGE) and could be used directly, whereas some others are codes (such as DIST) or dates (DLAST). So the first step was to recode the data to make them compliant with the use of numerical techniques. For example :

- DAN (Diagnosis Date) and DLAST (Date last seen alive or date of death) were combined to obtain Survival time (DLAST - DAN).
- DIST (Residence subregion) was discarded because there was no way to recode it in a sensible way for this study, though for geographic visualization studies this code could be used to display the results of any analysis; ID was discarded too as it was only an identifier.
- DHA (Residence) and IDCO-M (ICDO morphology code) could have been discarded because they are codes, but later we will see that they are meaningful.
- HADSURG (Did patient has surgery) values were rescaled from "1-No, 2-Yes,3-Laparotomy only" to "1-No,2-Laparotomy only, 3-Yes", as Laparotomy is a minor surgical procedure and hence it becomes a scaled ordered value.

Then, all the patients with a survival time lower than one week were dismissed, as they seemed to be extreme cases and would introduce a bias in the results.

2.2 Missing data problem

Medical data are notorious for their missing data. This dataset does not contradict that rule as we can see in Table 2.1 :

Variable	Number of missing data
ICDO-M	74
ICDO-B	74
STAGE	257
ADEQ	346
HISTO	141
GRADE	714
HADSURG	76
SURGEON	139
RESDIS	551
PREVHYST	403
OPTYPE	160
HADCT	944
TYPE	1333

Table 2.1: Number of missing data

To use a Neural Network, or almost any pattern processing structure, we need to have a complete vector of inputs. So we could either dismiss all the patients with at least one missing variable, or dismiss all the variables where at least one value was missing or finally try to fill in the missing data. We decided to dismiss all the 3 variables with more than half the values missing (GRADE, HADCT and TYPE), and then to fill in the other values using data imputation methods.

To fill in missing values in fields in the dataset, first a naive approach using the conditional probability has been tried using the following principle : the probability of an event 'a being α' is the sum of all the probabilities of the events 'a being α and b being β' times the probabilities of the events 'b being β'. So if we consider the event 'a being α' is the event 'variable a has the value α' and the event 'b being β' as 'variables b have the values β', it comes

$$p(a = \alpha | b = \beta) = \frac{p(a = \alpha) - \sum_{b \neq \beta} p(a = \alpha | b) p(b)}{p(b = \beta)} \tag{2.1}$$

where a is the missing value to estimate and b indexes other full columns of the matrix. Unfortunately, this leads to very poor results as the probabilities of b are small as b has the length of all the variables with all missing values, so nearly all the β occur only once in the dataset, so $p(a = \alpha | b = \beta)$ is nearly $p(a = \alpha)$.

Since the previous methods was not successful, another approach was considered. The missing data can be assumed to be values with infinite noise level. So we considered

using a novel method used in picture reconstruction which provides excellent results : the noisy ICA (Independent Component Analysis). This methods is performed to estimate the true values of the missing data[6] as following.

Let us call the original dataset D, its subset without any missing data $D_{complete}$ and let us assume it is composed of a linear superposition of basis function plus additive noise so we have:

$$D = A\hat{s} + \epsilon \tag{2.2}$$

where A is the $L * M$ mixing matrix of the squared ICA whose columns are the basis functions, $\hat{s}$ is a $M * N$ matrix of basis coefficients, and ϵ is the noise on the dataset. First we have to compute the matrices A and s as follows :

$$D_{complete} = As \tag{2.3}$$

as $D_{complete}$ has no noise on it due to missing data, since all values are known. (At this stage we are assuming that the collected data is accurate). Then we need to impute the matrix $\hat{s}$ of basis coefficients which maximizes the prior

$$\hat{s} = \max_{s} P(s|D, A) \tag{2.4}$$

Using Bayes rule, we have

$$P(s|D, A) = P(D|A, s)P(s) \tag{2.5}$$

The missing data can be viewed as a form of Gaussian noise. If the noise level on each coefficient is λ_i, the likelihood function has the form :

$$P(D|A, s) \propto exp(-\sum_{i} \frac{\lambda_i}{2}|D - As|_i^2) \tag{2.6}$$

where i is the index within the matrices. On missing data, we say that the variance of the noise is infinite, so $\lambda_{missing} = 0$ and on the other variables, the noise level is low so their λ is high.

The prior that has been used is to have the same mean for the features on each of the matrices s and $\hat{s}$:

$$P(s) \propto exp\left(-\sum_{i}(\frac{\sum_j(s_{i,j})}{N} - \mu_i)^2\right) \tag{2.7}$$

where μ_i is the mean of the i^{th} line of s. So taking the negative logarithm, the cost-function to optimize is :

$$\hat{s} = \min_{s}\left[\sum_{i} \frac{\lambda_i}{2}|D - As|_i^2 + (\frac{\sum_j(s_{i,j})}{N} - \mu_i)^2\right] \tag{2.8}$$

And finally the denoised (in the sense of missing data) dataset $D_{reconstructed}$ is computed as

$$D_{reconstructed} = A\hat{s}. \tag{2.9}$$

This reconstruction has been performed in two parts (see figure 2.1).

1. First of all, the data imputation has been performed using a subset $Dcomplete_1$ of D containing all data without the variables Survival Time which is the target, nor RESDIS (the residual decease) neither PREVHYST (previous hysterectomy), and without any missing data. This left a 875 * 26 matrix as a prior matrix to fill in (impute) only 896 missing values. Then s_1 has been computed by maximizing the cost-function (2.8) and given s_1 we obtained D_1, and the values of D_1 corresponding to the discrete variables have been rounded.

2. Then RESDIS and PREVHYST were added to D_1 to obtain $Dcomplete_2$. It provided a 705 * 28 matrix as a prior to fill in the 1700 missing values (of which more than a half have already been estimated in the first part, but are being re-estimated here). Then s_2 was computed as well as D_2 and round the discrete values of D_2. Finally we obtain $D_{reconstructed}$.

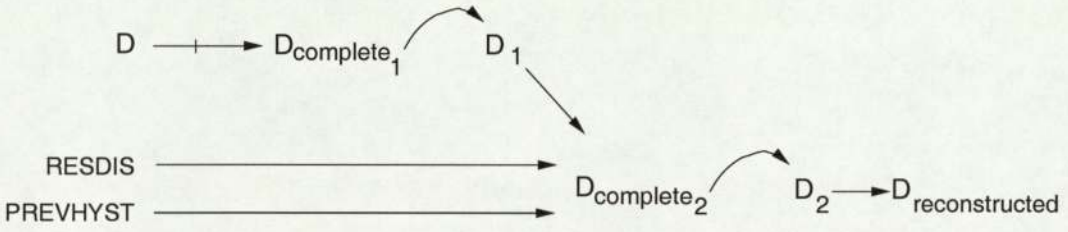

Figure 2.1: Overview of the method to fill in the missing data

2.3 Normalization

As the dataset is made of variables of different type and scale, some variables are likely to be given higher importance than they should only because of their scales rather than their global relevance. Indeed, while the values of ICDO-M (ICDO morphology code) are lying between 8000 and 9110, most of the others variables are going from 0 to 9. For each variable $\boldsymbol{X_i}$ we compute its mean $\overline{\boldsymbol{X_i}}$ and variance σ_i^2 to obtain $\widetilde{\boldsymbol{X_i}}$, the normalized variable with zero mean and unit standard deviation

$$\widetilde{\boldsymbol{X_i}} = \frac{\boldsymbol{X_i} - \overline{\boldsymbol{X_i}}}{\sigma_i} \tag{2.10}$$

Note that it is a simple linear transformation, so for a MLP[1] it is not a such important issue since unnormalized data can be handled by simply changing the initialization methods as the first layer of the network can normalize the data. On the other hand, this is a critical issue for the RBFs Networks[2] as the activation of the basis function are determined by the Euclidean distance between the input vector and the basis function center.

2.4 Splitting the dataset

For later reference, we also divide the dataset $D_{reconstructed}$ into two components :

- $D_{censored}$ which contains only the data referring to censored patients ($STAT = 1$),
- D_{dead} which contains the data referring to patients with deadly outcome ($STAT = 2$).

[1] See 4.2.2
[2] See 4.2.3

Chapter 3

Lowering the dimensionality : the Variable Selection

Once the imputation of missing data is completed, it is useful to check whether all the variables are relevant to the task of prognosis or not, so we can reduce the dimensionality of the inputs of the Neural Networks or other machine intelligence approaches. The variable selection is a good way to keep only the most relevant variables of the dataset. Lowering the dimension of the data avoids a too complex model by removing the redundant or irrelevant information. Contrary to the feature selection which is a transformation of all the variables to obtain features in lower dimensions, the variable selection keeps only the interesting variables using different criteria. In this part, two different methods of variable selection will be introduced : the Mutual Information[4] and the Automatic Relevance Determination[2].

Lowering the number of inputs is a good way to prevent a too high complexity of the neural network, and it is better to extract only the relevant information about the patient.

3.1 Mutual Information

The mutual information is a measure of information contained between two variables, defined as the difference of the entropy of one of the variables and the cross entropy between the variables :

$$I(X,Y) = \sum_{x,y} p(x,y) \ln \frac{p(x,y)}{p(x)p(y)} = H(X) - H(X|Y) \tag{3.1}$$

Figure 3.1 shows the calculated mutual information between all of the variables of the reconstructed dataset $D_{reconstructed}$. The Survival Time, our desired target, is the 29^{th} variable on the extreme right-hand side of the plot and we can see it has a high mutual

information with several variables listed in Table 3.1. The threshold selected is 1, since the values below 1 are relatively lower.

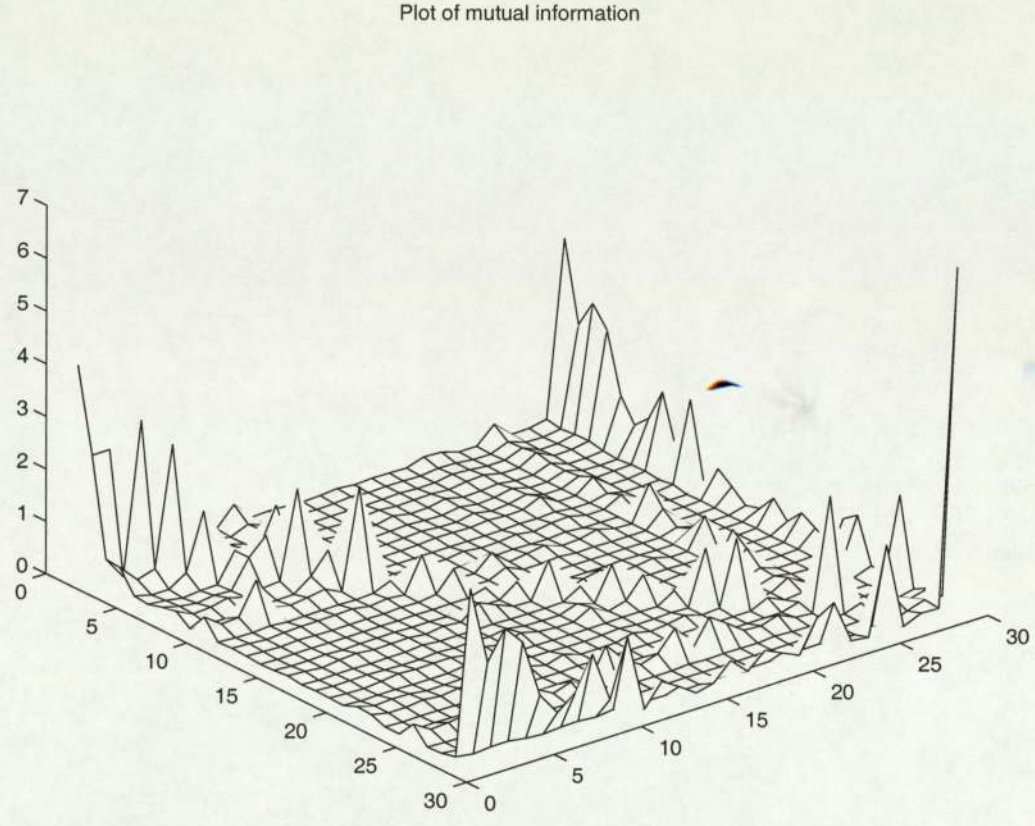

Figure 3.1: Plot of the mutual information

NAME	VALUE
ICDO-B	1.0966
PREVHYST	1.2813
STAGE	1.5818
HISTO	1.5856
OPTYPE	1.9397
AGP	1.9414
ICDO-M	2.1123
DHA	2.4706
AGE	3.4436
SURVIVAL TIME	6.5995

Table 3.1: Highest values of the mutual information with the survival time

These results are quite logical since nearly all these variables are related to the tumor and are commonly used by doctors to set prognosis, except for DHA (Residence) which tends to suggest that the medical procedures are more efficient in some regions than

other. So using this threshold, the Mutual Information method selected 9 variables from the 27 which could be used.

To check these results for consistency, an Automatic Relevance Determination has also been carried out.

3.2 Automatic Relevance Determination

3.2.1 Introduction

Automatic relevance determination[2] is a Bayesian technique based on the *evidence* framework introduced by MacKay in [8]. Each input is associated with a hyper-parameter α, which can be regarded as the variance of the network with respect to the input it is associated with. Two ARD experiments had been performed : one for the regression case, and one for the classification case, as it is not obvious that the same inputs would be relevant.

3.2.2 The regression case

An ARD has been performed with all the dataset $D_{reconstructed}$ using a MLP (see Section 4.2.2), using all the variables as inputs and the Survival Time as target, and we can see in Figure 3.2 that some variables are completely irrelevant, those with the highest values of α whereas others are not, those with a low value of α. It ended with the selection of the 15 variables listed in table 3.2, using a value of 50 for α as threshold, mainly to make sure we were not taking away some variables which might be useful.

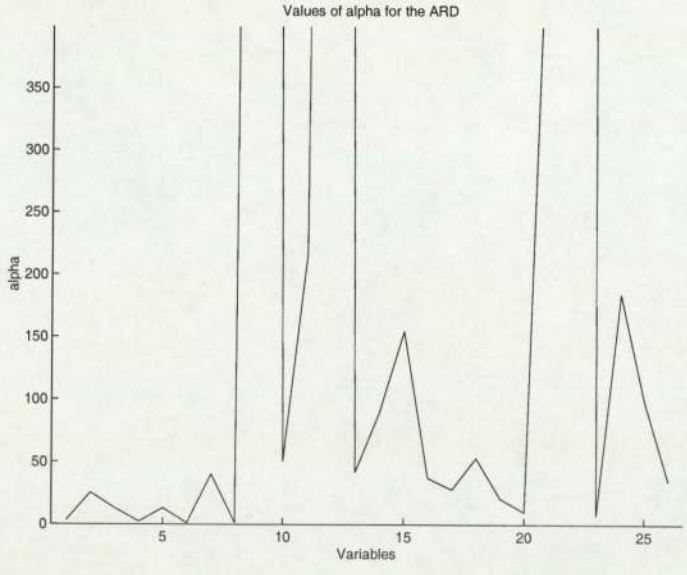

Figure 3.2: Plot of the values of α for each inputs of the neural network

NAME	VALUE
STAGE	0.9
HISTO	1.3
ICDO-M	2.1
AGE	3.3
IDS	7.3
SURGEON	9.9
ICDO-B	12.9
DHA	13.0
NODES	20.8
AGP	25.3
OMENT	27.9
OTMALIG	34.5
LAVAGE	37.3
ADEQ	40.2
SUBTAH	41.9

Table 3.2: Lowest values of α

3.2.3 The classification approach

Here, ARD has been performed with the whole dataset using also a MLP (see Section 4.3.2) with all the variables as input, and the class where the patients belongs as output(see Section 4.3). The results are not exactly the same as in the regression case, as shown on Figure 3.3. The name of the most relevant inputs are given in Table 3.3 together with the corresponding values of the α parameter. In this case, the values of the α parameter are much lower, so a lower threshold was used, because otherwise all values but one would have been used. Several attempts of training classifiers with different number of inputs show that the best threshold to be used was 7.5, ending with the selection of 20 variables.

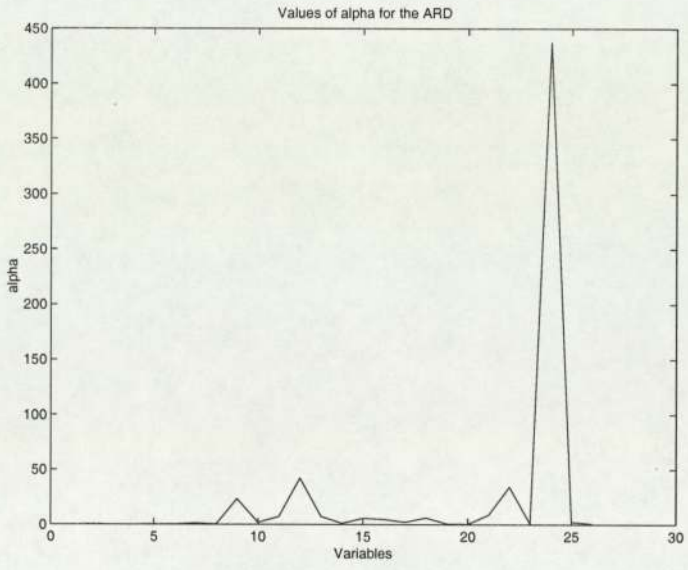

Figure 3.3: Plot of the values of α for each inputs of the neural network

NAME	VALUE
AGE	0.0398
ICDO-M	0.0692
DHA	0.0990
HISTO	0.2485
STAGE	0.2543
AGP	0.2591
RESDIS	0.3647
ICDO-B	0.4123
INTERVAL	0.6611
SURGEON	0.6914
OPTYPE	0.7835
BSO	1.1183
ADEQ	1.1814
TAH	1.8231
BIOPSY	2.1905
OTMALIG	2.5306
OMENT	4.7397
LAVAGE	5.7153
NODES	6.1147
OOPH	7.1980

Table 3.3: Lowest values of α in the classification case

The results of both ARD are more or less the same, with some extra variables selected for the classification approach. Surprisingly, the ARD in the regression approach did not select the variable RESDIS (Residual Disease) while it is known that the size

of tumor after an operation is a factor of survival[3]. This might be explained by the noise embedded in this variable as it is known that some surgeons over-estimate the size while others underestimate it, such that the α parameter linked with this input becomes too big, in the regression case.

3.3 Conclusion

The results of both methods have been combined by keeping the variables retained by both the Mutual Information and the ARD, such that we end up with 15 variables as input in the regression approach, which is a reduction by nearly a half of the initial data dimensionality. On the other hand, in the classification approach 21 variables are retained, only a reduction by a quarter of the number of variables.

Chapter 4

Neural Network Models

Neural Networks are useful prediction tools, especially when we do not know the mapping function between the input and the target, either because it is too complicated, or because we have no hints of what it is like. Neural Networks are able to fit nearly any function, with any given precision[2], by feeding it the inputs, and by optimizing its parameters such that the outputs correspond to the desired ones : this is called training.

As most of the standard methods are unable to provide good results on this kind of problem, Neural Networks represent an interesting and promising approach.

4.1 Training method

As said earlier, Neural Networks are able to fit nearly any function, but we want to have good estimation not only on the dataset we have, but also to generalize the result to new patients. This is why we split the dataset in two equal parts : the *training set* and the *test set*. The training set was used to train the Neural Networks while the test set was used to check whether the generalization was good. Indeed, the main problem with Neural Networks is *overfitting*, ie the network is trained too much and fits the data of the training set and its noise perfectly, but has very poor results on other data. The method we used for the training, is the *hold out* method which consists in stopping the training when the error on the test set, the test error, starts to increase as this implies that the Neural Network stops learning the underlying function and begins to learn and fit the noise as well. Hence the test error is a good measure of the generalization of the Network since the noise component has different values on both sets.

4.2 The regression approach

In this approach, the goal is to predict how long a patient will survive as an analogue value of the survival time. Hence the input of the Neural Networks are the variables that have been selected earlier, whereas the output is the Survival Time, our target.

4.2.1 Error measure

The error measure that is used for the regression approach is the normalised relative mean squared error

$$Err = \sqrt{\frac{\sum(y_i - t_i)^2}{\sum(t_i - \bar{t})^2}} \tag{4.1}$$

where y_i and t_i are the network output and the target (Survival Time) for the i^{th} pattern, and $\bar{t}$ is the mean of all the targets, which is the average Survival Time. An error value of 0 means a perfect fitting, whereas a value of 1 means randomness.

4.2.2 Standard point estimators : the Multilayer Perceptron

The Multilayer Perceptron[2] (MLP) is a feed forward neural network composed of several hidden layers of adaptive weights.(Figure 4.1)

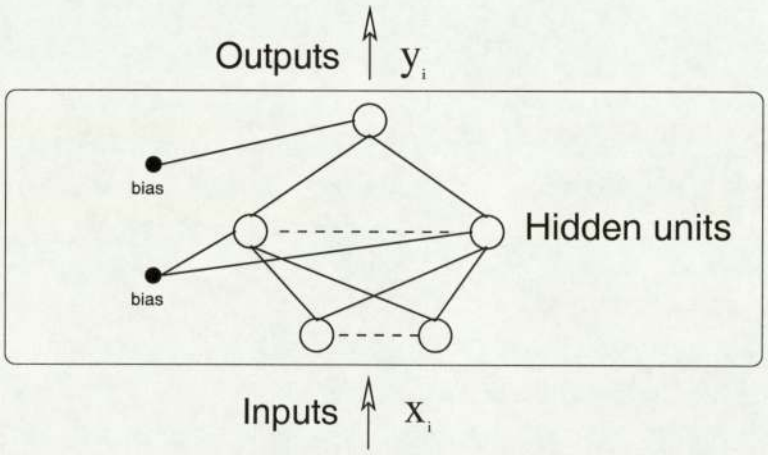

Figure 4.1: Multilayer Perceptron

The inputs are propagated through each layer of the network, according to the weights and activations functions used, and finally to become the output. The value v_i^{h+1} of the neuron i of layer $h+1$ is

$$v_i^{h+1} = f\left(\sum_{j=1}^{N_h} w_{ij}^{h,h+1} v_j^h + b_i^h\right) \tag{4.2}$$

where ^{h} is the layer index, N_h is the number of neurons in layer h, b_i^h is the threshold of layer h for the i^{th} neuron, $w^{h,h+1}$ is the weight matrix between layers h and $h+1$ and f is the activation function.

To build a network which leads to an output from a given set of inputs, it is necessary to adapt the weights and biases. So we introduce the likelihood of the dataset

$$L = \prod_q p(t^q, x^q) = \prod_q p(t^q|x^q)p(x^q) \tag{4.3}$$

and for the computation, we introduce the error function E as the negative log-likelihood

$$E = -lnL \tag{4.4}$$

The standard MLP

If we consider the distribution of the dataset to be Gaussian, we have

$$p(t_k|x) = \frac{1}{(2\pi)^{\frac{1}{2}}\sigma} exp\left\{-\frac{(F_k(x) - t_k)^2}{2\sigma^2}\right\} \tag{4.5}$$

where $F_k(x)$ is the underlying generator function of the dataset. Using this distribution of the data, as well as (4.3) and (4.4), it comes:

$$E = nc\ln\sigma + \frac{nc}{2}\ln(2\pi) + \frac{1}{2\sigma^2}\sum_q\sum_k(f_k(x^q;w) - t_k^q)^2 + \sum_q \ln p(x^q) \tag{4.6}$$

where $f_k(x^q;w)$is the approximation by the neural networks of $F_k(x)$, w are the weights of the network, n the number of samples and c the dimension of the output. In this equation, only the middle part is a function of the networks, that is why the cost function to be minimized is :

$$E = \frac{1}{2}\sum_{q=1}^{n}\sum_{k=1}^{c}\left(v_k^{out,q} - t_k^q\right)^2 \tag{4.7}$$

where q is the training sample index, k is the output vector index, and t^q is the desired output vector for the q^{th} input.

Given the feed forward nature of the network and the fact that the activation function is differentiable, the derivative of this error function in respect of the weights and biases of the network can be found. This enables the training of the network using optimization algorithms in order to adapt the weights and biases so a given set of inputs leads to a corresponding output[2].

As we can see in Figure 4.2, the results are better than random but not really good enough to use in an advisory environment, with the normalized RMSE for the test set always greater than 0.79. The scatter-plots 4.3 and 4.4 of target versus output show that the network overestimates the survival time of patients with low survival time while it underestimates those with high actual survival time.

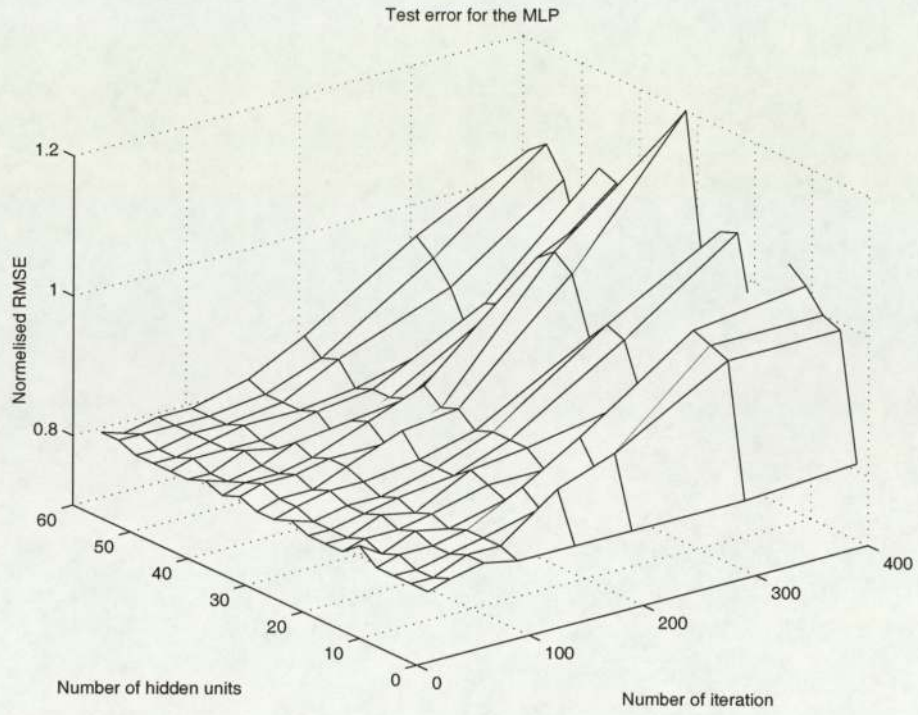

Figure 4.2: Test error for the MLP, the minimum test error is 0.79 reached with 5 hidden units and 20 iterations

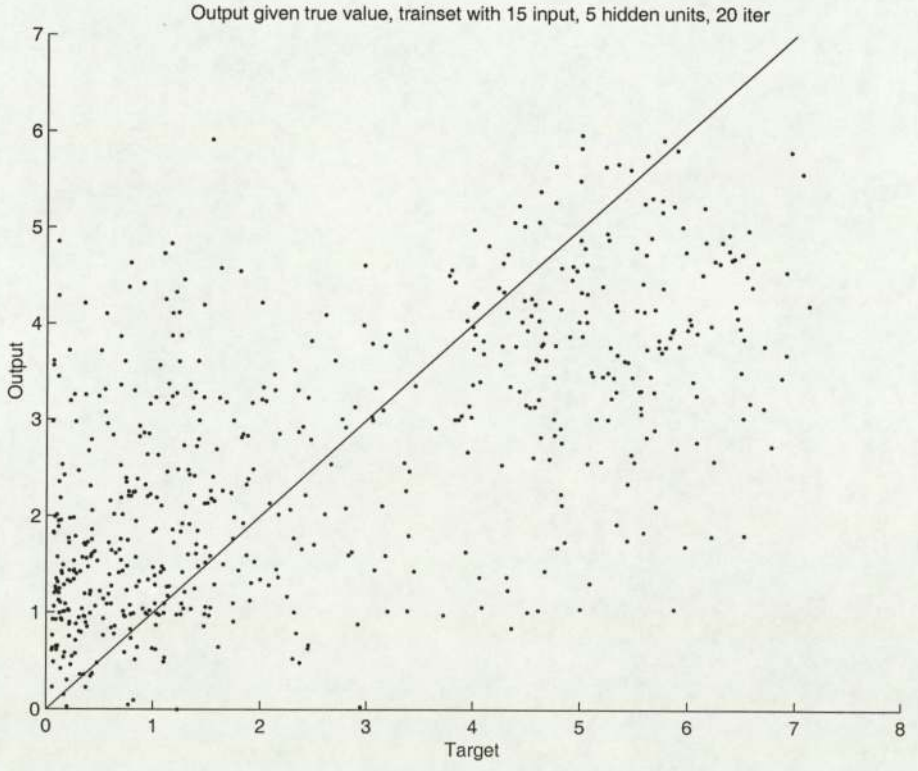

Figure 4.3: Plot of the target vs output for a standard MLP on the train set

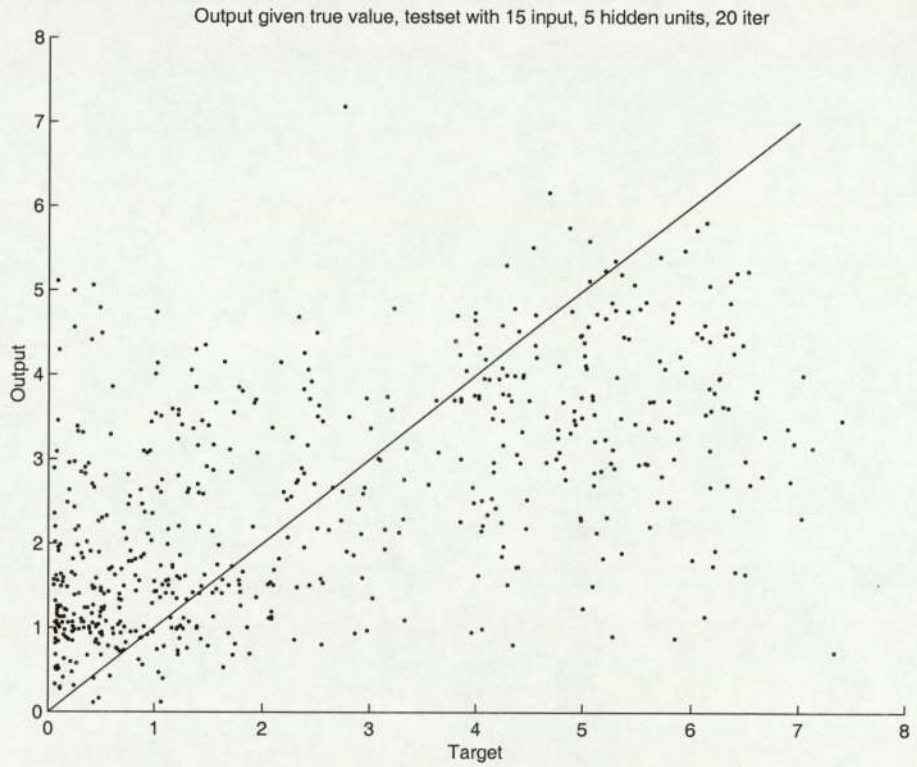

Figure 4.4: Plot of the target vs output for the standard MLP on the test set

MLP with a censored-compliant cost-function

One problem with this simple approach is that the MLP is trying to fit the survival time for each of the patients, but the actual survival time for the censored patients is not known as there was no follow-up study to termination. Therefore we considered introducing a different type of error function using this prior knowledge which should be able to cope better with this censoring[9] :

$$e = \begin{cases} (max(target - predicted, 0))^2 & D_{censored} \\ (predicted - target)^2 & D_{dead} \end{cases}$$

The error measure we used for this network was slightly different, as this MLP is made to overestimate the survival time of censored patients. Hence we do not take the overestimation of survival time for censored patients in the RMSE into account.

$$Err = \sqrt{\frac{\sum_{i \subset D_{dead}} (y_i - t_i)^2 + \sum_{i \subset D_{censored}} (max((t_i - y_i), 0))^2}{\sum (t_i - \bar{t})^2}} \tag{4.8}$$

The results for the MLP with an error function able to cope with censored data are in Figure 4.5. We can see that the results are slightly better with a RMSE of only 70%, but the scatter-plots of target versus output on Figures 4.6 and 4.7 are not fundamentally different from those of the standard MLP.

It can be assumed that the improvement in the error measure is mainly due to the fact that the overestimation of the survival time of censored patients is not taken into account. Hence the overestimation which was previously considered as an error is not any more, and the modification of the cost-function offers only slight improvement of the model.

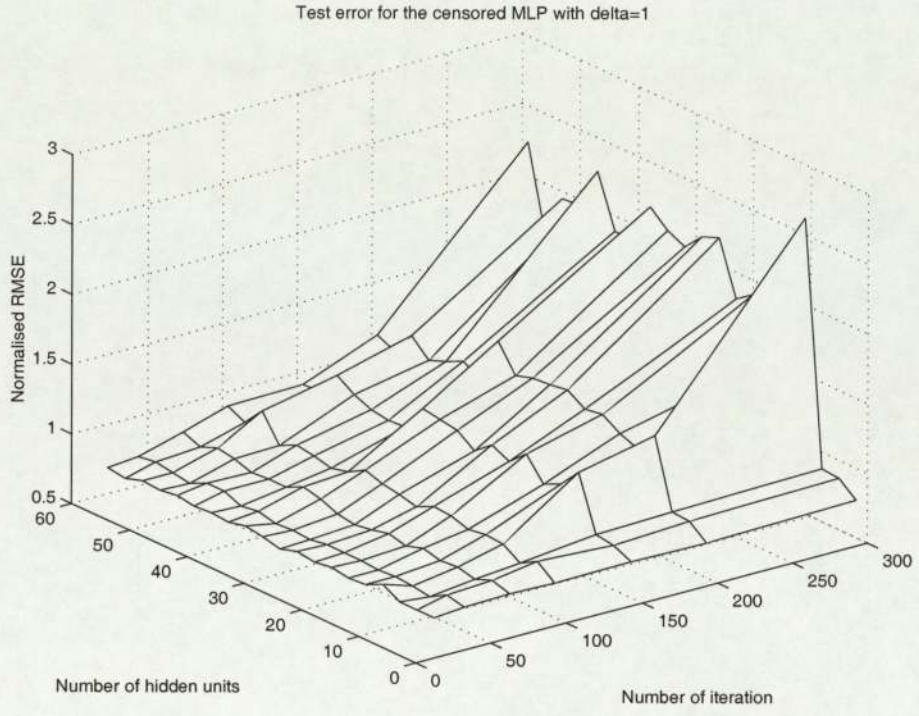

Figure 4.5: Test error for the MLP with the censored compliant cost-function, the minimum test error is 0.70 reached with 56 hidden units and 20 iterations

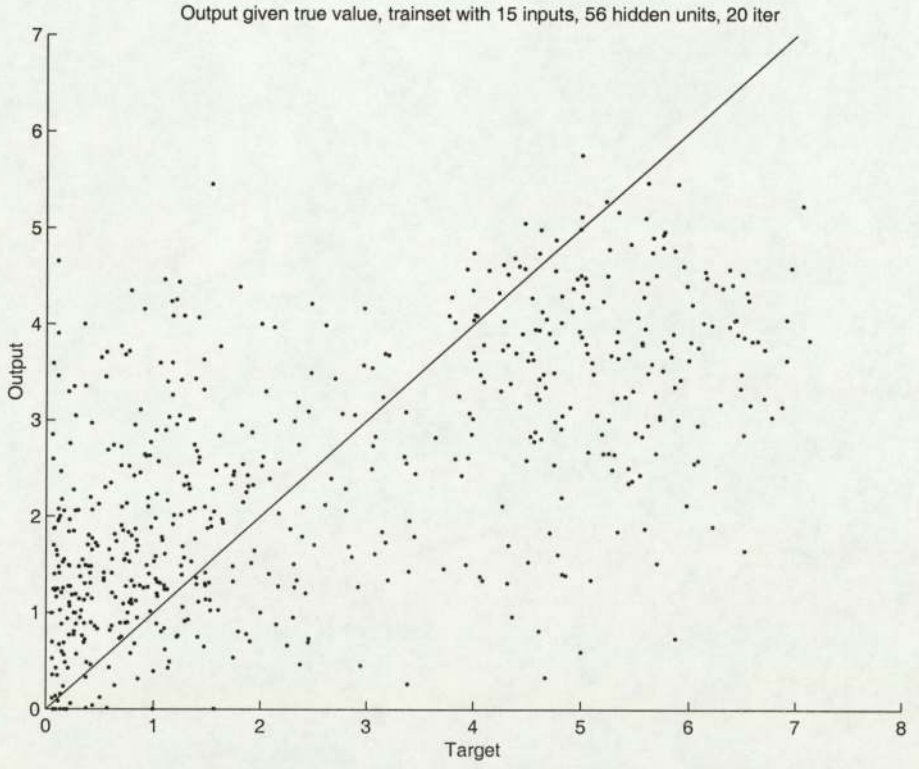

Figure 4.6: Plot of the target vs output for the MLP with the censored compliant cost-function on the train set

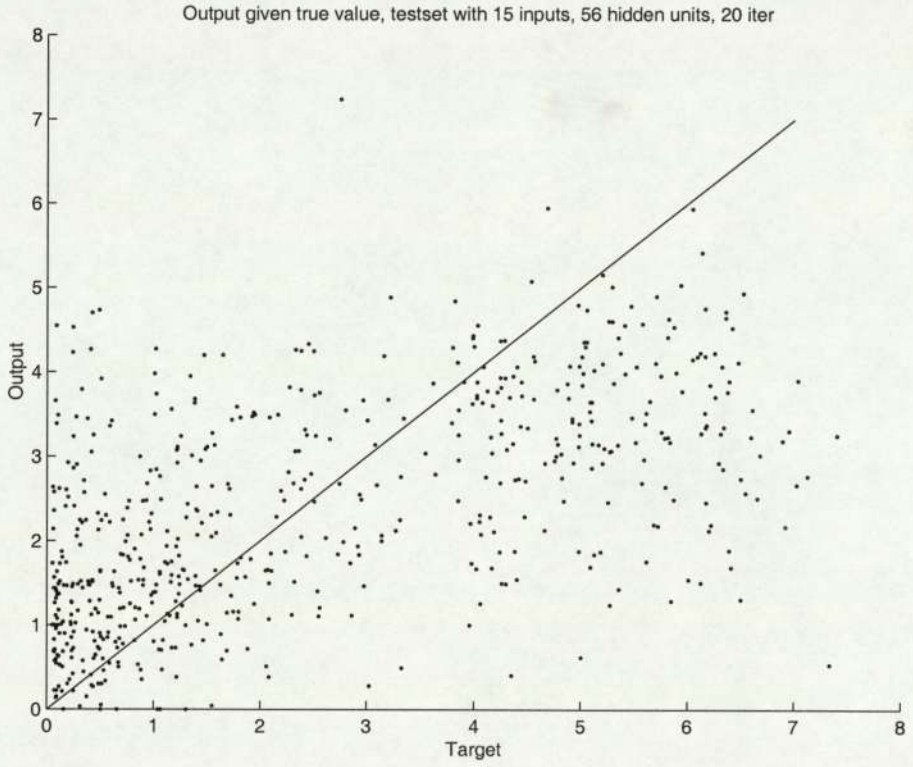

Figure 4.7: Plot of the target vs output for the MLP with the censored compliant cost-function on the test set

MLP with exponential cost-function

One of the main issues with the sum-of-square error is that it receives the largest contributions from the data with the largest errors. If the distribution has long tails or if some of the data are mislabeled, the solution is dominated by only a finite number of points called *outliers* (see Figure 4.8)

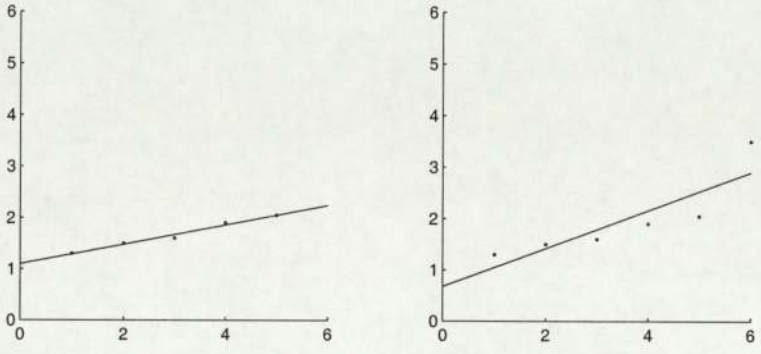

Figure 4.8: Example of fitting a linear polynomial through a set of noisy points with a sum-of-square cost-function. On the left, there is a good fitting, whereas on the right, with one extra point away from the others, the fitting is dominated by this point

For the rest of this thesis, let us call *funcexp* the continuous function defined as :

$$funcexp(x) = \begin{cases} -\beta x & \text{if } x \leq -\epsilon \\ x & \text{if } x \geq \epsilon \\ \sum_{i=1}^{7} \kappa_i x^i & \text{if } -\epsilon \leq x \leq \epsilon \end{cases}$$

The 7^{th} order polynomial between $-\epsilon$ and ϵ is to make the function continuous and differentiable to the 2^{th} order, and β is a parameter to allow stronger penalties on underestimated values as we know the minimum survival time of each patient. Indeed, for each patient we only know the diagnosis date and the date when the patient was last seen alive. So the actual survival time is longer than the one we know, and a right-sided model is interesting as it allows stronger penalties on the underestimation of the Survival Time.

In this part, the negative log-likelihood of cost function to be optimized is :

$$E = \sum_q \sum_k funcexp(\tau_k^{out,q} - t_k^q) \tag{4.9}$$

The Table 4.1 shows the minimum test error obtained for different values of β. The best results are for $\beta = 1$ so skewing the distribution in this case is not interesting. The scatter-plots of target versus output on Figures 4.9 and 4.10 are a bit worse than before, as the spread of the points is wider.

Value of β	1	1.1	1.25	1.5	1.75	2	2.5	3
Min test error	0.80	0.82	0.82	0.81	0.82	0.86	0.87	0.90

Table 4.1: Minimum values of the test set error for the MLP with exponential cost-function given β

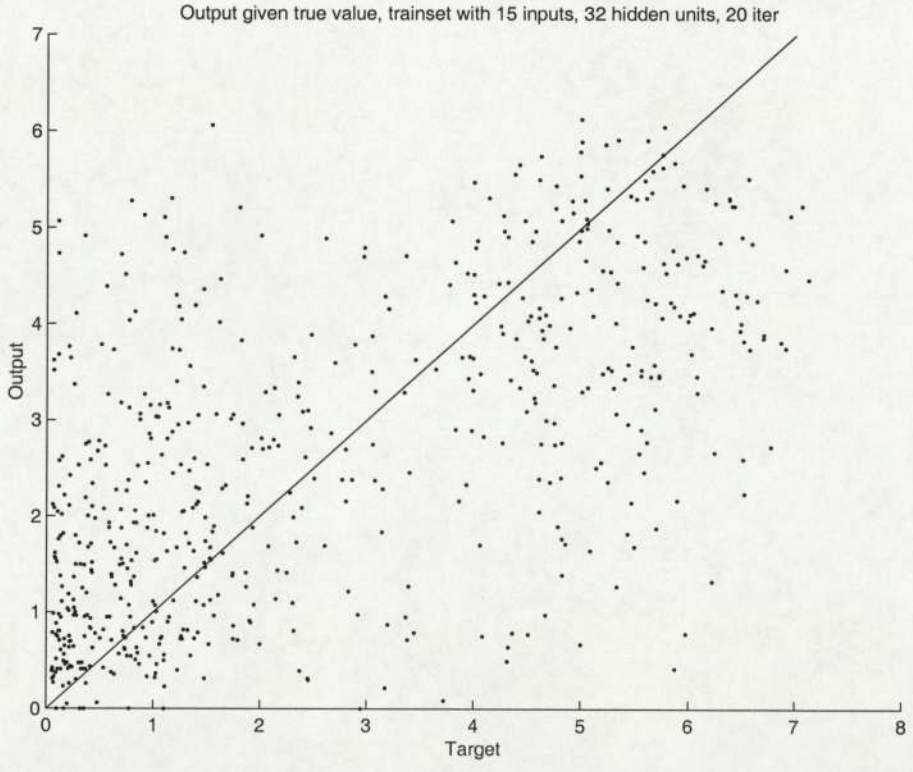

Figure 4.9: Plot of the target vs output for the MLP with exponential cost-function for the train set

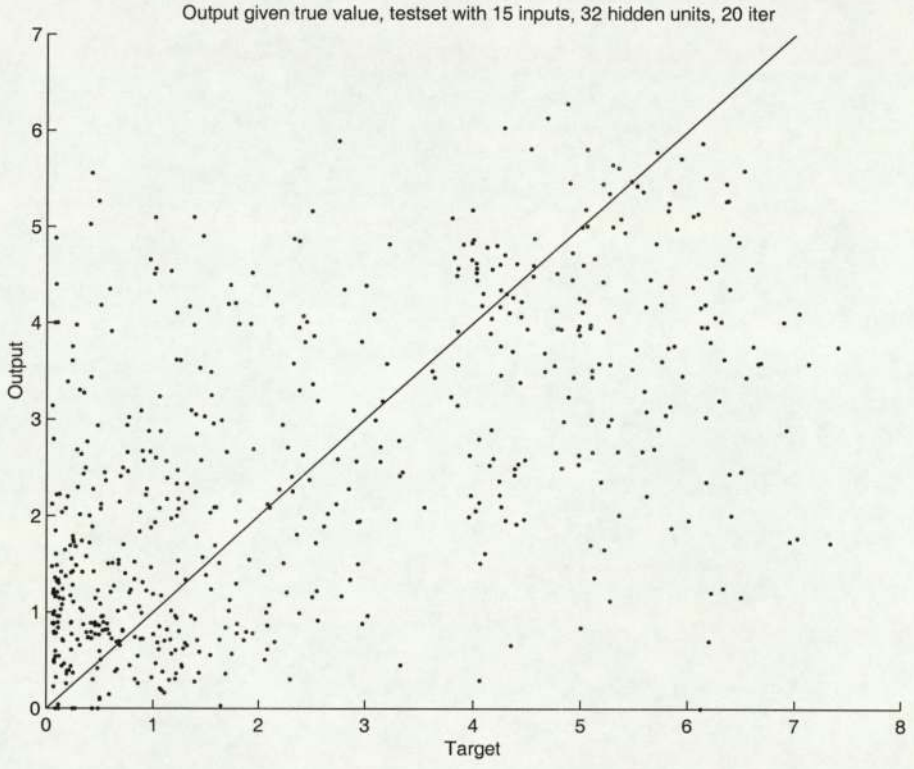

Figure 4.10: Plot of the target vs output for the MLP with exponential cost-function for the test set

None of the attempts to improve the results of the MLPs by modifying the cost-function and the noise models assumptions has been successful, so another kind of network has been tried: the Radial Basis Function network.

4.2.3 Standard point estimators : RBF networks

Radial Basis Function networks[11] (RBF) are related to kernel methods for density estimation and regression, and to normal mixture models. The idea of a RBF model is to expand a given function f using a set of basis function of the form $\Phi(\| x - x^n \|)$, where Φ is a non-linear function to be chosen. The output is then taken to be a linear combination of the basis functions :

$$f(x) = \sum_n w_n \Phi(\| x - x^n \|) + w_0 \tag{4.10}$$

where w_n is the weight of the n^{th} basis function and w_0 is the bias. Several forms of basis function can be used, a common one is the thin-plate spline:

$$\Phi(x) = x^2 \ln(x) \tag{4.11}$$

which is the best solution for curve fitting.

A radial basis function network uses several RBFs as hidden units. (Figure 4.11)

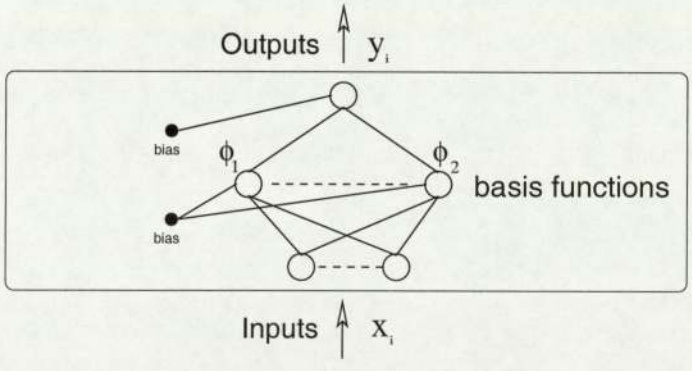

Figure 4.11: RBF network

The interpolation formula 4.10 is then:

$$y_k(x) = \sum_{j=1}^{M} w_{kj} \Phi_j(x) + w_{k0} \tag{4.12}$$

and the thin-plate spline basis function can be expressed as:

$$\Phi(x) = \| x - \mu_j \|^2 \ln(\| x - \mu_j \|) \tag{4.13}$$

where x is the input vector and μ_j is the vector determining the centers of the basis function Φ_j. Once the basis function have been chosen, we have a simple model. Its parameters can be found by a least squares, or any other optimization procedure.

For a large class of basis functions, RBF networks are universal approximators[11]. Besides, they possess the property of best approximation, which means that the set of functions corresponding to all possible choices of the adjustable parameters includes the optimal approximation. One advantage of this family of networks is that RBF models are very fast to train in comparison to networks with sigmoidal units.

Table 4.2 shows that the results are more or less the same as those with the standard MLP. The plots of target vs output on Figures 4.12 and 4.13 are very much like those for the standard MLPs and those with a censored compliant cost-function. It can be concluded that standard Neural Networks are unable to provide good generalization with this dataset.

	Number of iterations								
Number of hidden units	30	60	100	150	200	300	400	600	800
16	47.10	9.41	1.63	0.85	0.81	0.81	0.81	0.81	0.81
32	35.35	7.60	1.76	0.99	0.82	0.83	0.83	0.96	0.96
64	34.53	13.01	3.76	1.36	1.02	0.84	0.89	0.95	1.03
96	164.24	21.14	2.41	1.13	0.85	0.82	0.84	0.86	0.86
128	176.60	36.70	7.46	1.46	0.94	0.83	0.85	0.85	0.85

Table 4.2: Test error for the RBF

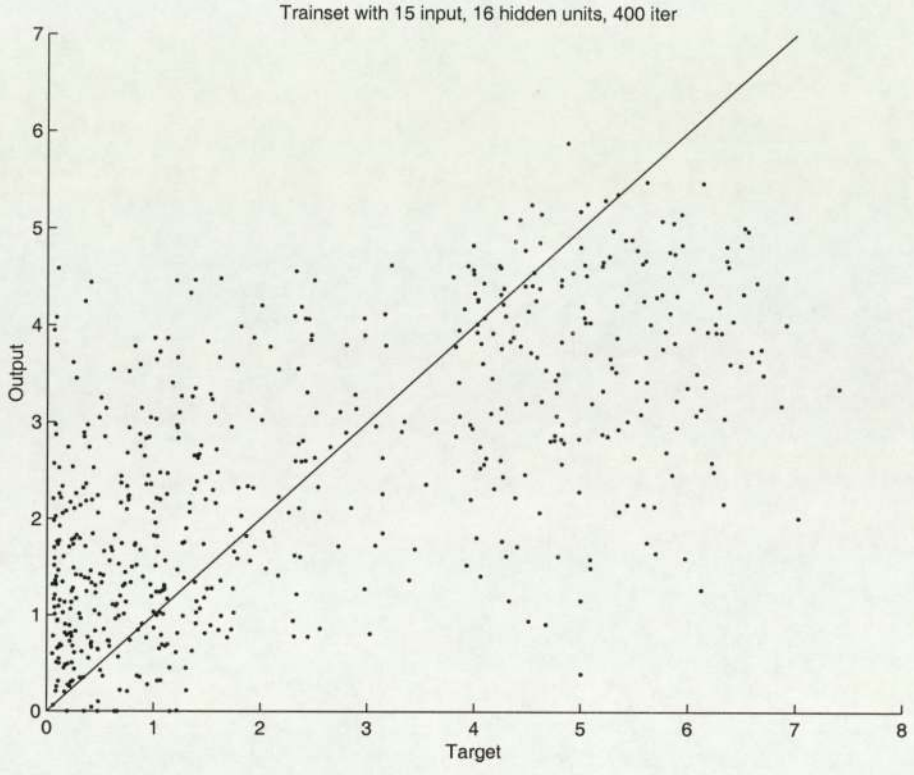

Figure 4.12: Plot of the target vs output for the RBF for the train set

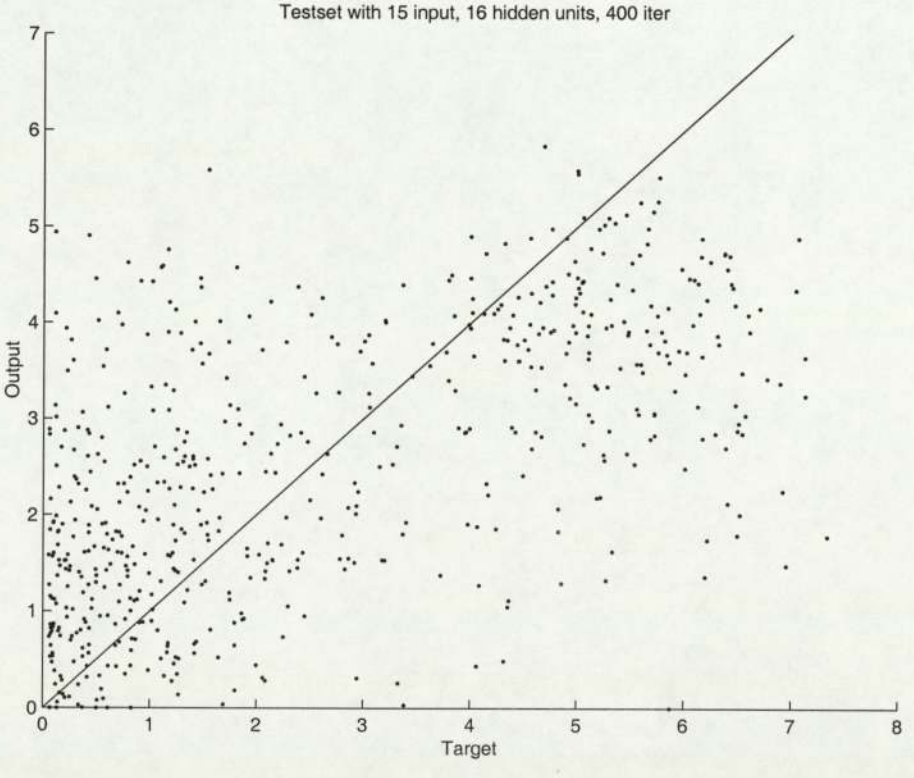

Figure 4.13: Plot of the target vs output for the RBF for the test set

4.2.4 The full distribution of the patients' prognosis : MDNs

So far, the results have been quite disappointing, with always more than 0.7 normalized error in the prediction of the survival time. Rather than trying a point fit, it could be better to use models which fit the whole distribution, and moreover these would provide confidence intervals, the error bars. So another approach was considered using the Mixture Density Networks[1] which combine a standard Neural Network with a mixture density model to provide a conditional distribution rather than a single output.

In other words we accept that patient prognosis for ovarian cancer is strongly stochastic and a simple point estimator for survival is not an adequate statistic to describe a given patients' prognosis. Instead, we need to have a better characterisation of the full distribution description the patients' prognosis.

Introduction

We consider the probability density of the target as a combination of kernel functions of the form

$$p(t|x) = \sum_{i=1}^{m} \alpha_i(x)\phi_i(t|x) \tag{4.14}$$

where m is the number of components in the mixture, the $\alpha_i(x)$ are the mixing coefficients (priors probabilities) and the $\phi_i(t|x)$ are the conditional densities of the target vector t for the i^{th} kernel. The implementation of such a model is straightforward as shown in Figure 4.14:

- First there is a Neural Network with an input vector $\mathbf{x}$ and an output vector $\mathbf{z}$ of parameters for the functions : priors (z^{α}), centers(z^{μ}) and variance/skewness($z^{\sigma/\lambda}$). So if c is the dimension of the target, the dimension of $\mathbf{z}$ is $(c+2)*m$: m for the priors, $c*m$ for the centers or origins of the function and m for their variances or skewnesses.

- Then there is a mixture model with the parameter vector $\mathbf{z}$ as input and $\boldsymbol{p(t|x)}$ as an output.

As this is a mixture model, there is the constraint

$$\sum_{i=1}^{m} \alpha_i(x) = 1 \tag{4.15}$$

This can be obtained by considering $\alpha_i(x)$ as the *softmax* function of the output of the neural network

$$\alpha_i = \frac{exp(z_i^{\alpha})}{\sum_{j=1}^{m} exp(z_j^{\alpha})} \tag{4.16}$$

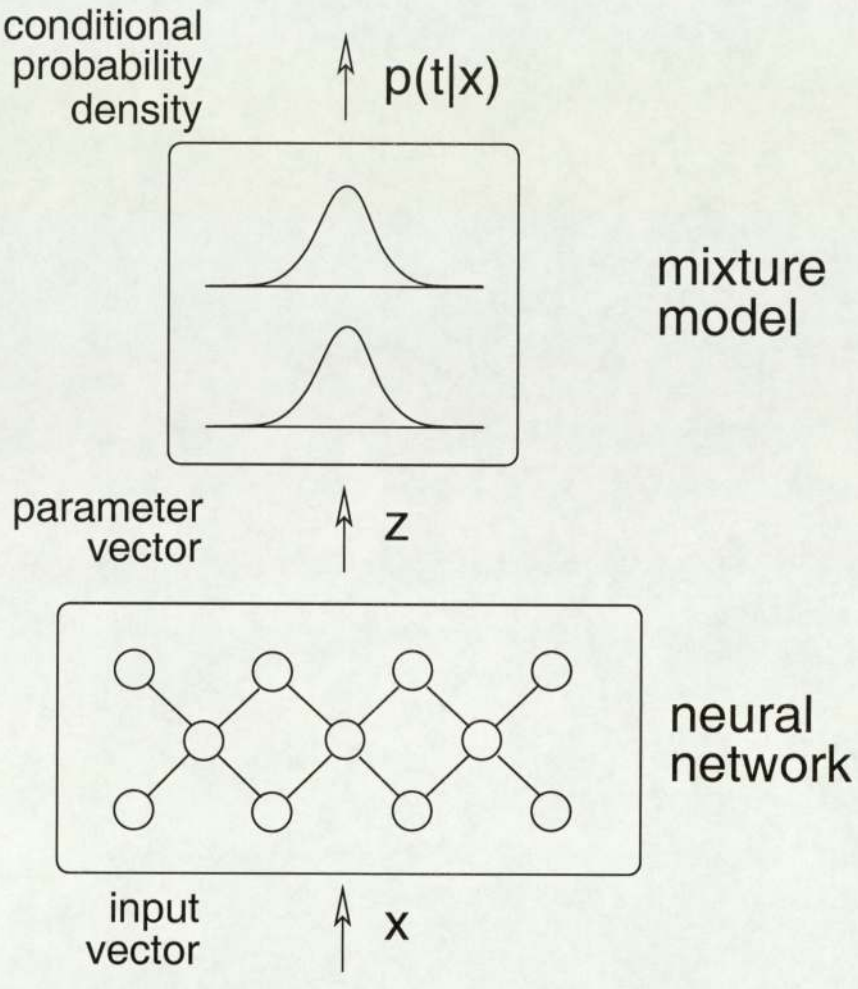

Figure 4.14: The Mixture Density Network

The centers/origins are simply the network outputs

$$\mu_{ik} = z_{ik}^{\mu} \tag{4.17}$$

Let us define the error function for the MDN as

$$E = \sum_{q} E^{q} \tag{4.18}$$

where the error from pattern q is the negative logarithm of the likelihood (4.3), without the terms $p(x^q)$ as they are independent of the parameters of the mixture model.

$$E^{q} = -ln \left\{ \sum_{i=1}^{m} \alpha_i(x^q) \phi_i(t^q|x^q) \right\} \tag{4.19}$$

To optimize the network, we have to minimize this function with respect to the output of the Neural Networks and then to back-propagate the modification of the output into

the MLP to optimize its weight. To simplify the computation, let us introduce the *posterior* probabilities, obtained using Bayes rule

$$\pi_i(x,t) = \frac{\alpha_i \phi_i}{\sum_{j=1}^{m} \alpha_j \phi_j} \tag{4.20}$$

Using (4.20) and (4.19) the required gradients can be calculated as :

$$\frac{\partial E^q}{\partial \alpha_i} = -\frac{\pi_i}{\alpha_i} \tag{4.21}$$

and using (4.16)

$$\frac{\partial \alpha_i}{\partial z_k^\alpha} = \delta_{ik}\alpha_i - \alpha_i \alpha_k \tag{4.22}$$

Using the chain rule :

$$\frac{\partial E^q}{\partial z_k^\alpha} = \sum_i \frac{\partial E^q}{\partial \alpha_i} \frac{\partial \alpha_i}{\partial z_k^\alpha} \tag{4.23}$$

So finally using (4.21), (4.22) and (4.23) it follows

$$\frac{\partial E^q}{\partial z_k^\alpha} = \alpha_k - \pi_k \tag{4.24}$$

The other gradients depend on the used distribution.

Gaussian kernels

First of all, the functions used as kernels are Gaussian (universal estimators[11]). In this particular case, the kernel function is :

$$\phi_i(t|x) = \frac{1}{(2\pi)^{c/2}\sigma_i(x)^c} \exp - \frac{||t - \mu_i(x)||^2}{2\sigma_i(x)^2} \tag{4.25}$$

where $\mu_i(x)$ and $\sigma_i(x)$ are the center and variance of the i^{th} kernel.
The variance parameter has to be always positive, so it can be considered as an exponential

$$\sigma_i = exp(z_i^\sigma) \tag{4.26}$$

Using (4.19), (4.20) and (4.25) the gradient with respect to σ_i is

$$\frac{\partial E^q}{\partial \sigma_i} = -\pi_i \left\{ \frac{||t - \mu_i(x)||^2}{\sigma_i^3} - \frac{c}{\sigma_i} \right\} \tag{4.27}$$

And as

$$\frac{\partial \sigma_i}{\partial z_i^\sigma} = \sigma_i \tag{4.28}$$

we have

$$\frac{\partial E^q}{\partial z_i^\sigma} = -\pi_i \left\{ \frac{||t - \mu_i(x)||^2}{\sigma_i^2} - c \right\} \tag{4.29}$$

And finally using to (4.20) and (4.25)

$$\frac{\partial E^q}{\partial z_{ik}^{\mu}} = \pi_i \left\{ \frac{(\mu_{ik} - t_k)}{\sigma_i^2} \right\} \tag{4.30}$$

Now we have the derivative of the cost-function with respect to all the outputs of the network.

The output of the MDN with Gaussian kernels is directly converted into a prognosis by taking the center of the Gaussian with the highest prior, as to say the most likely Survival Time. Another approach was considered using the mode of the whole distribution (4.14) but the results obtained were worse than those with only the mode of the Gaussian kernel with the highest prior.

The variance of this Gaussian provides a confidence level in the output.In Figures 4.15, 4.16 and 4.17 we cannot see any major improvement in the results. The results with more than 6 kernel functions are not shown as they are as bad, or worse than those with 6 kernels.

Even worse, the best result for the test set is for 2 Gaussian kernels and has a normalised error of 0.82. The scatter-plots of target versus output in Figure 4.18 and 4.19 are very similar to all the previous results. This can be easily explained by the fact that the plots are representing the mode of the distributions and by looking closely to the central plots on Figures 4.20 and 4.21, we see that it is always the same gaussian kernels which has the highest prior, so all the assets of using a MDN is lost, except for the fact that we have the variance on each output and so we have a confidence interval. Indeed, the third plots on Figures 4.20 and 4.21 show the values of the variances of the Gaussians given the actual survival time. The curve of the variances of the only Gaussian which is being used shows that its variance is quite low, about 0.5 year, resulting in an fairly good confidence in the results, even if the plots of the target versus output are showing that the MDN is nearly always missing the target, and has not learned the underlying parameter.

The plots 4.22 and 4.23 show the full distribution $p(t|x)$ as a density. They represent the probability for a patient on the x-axis to have its Survival Time with the same value as the Survival Time indexed on the y-axis. The patients have been ordered by Survival Time, so the aim is to obtain a diagonal, which means that the MDN can fit the target values (the Survival Time). For every patient, the output probabilities have been divided by its maximum value to obtain a normalised plot, therefore we can see the distribution for each patient without changing the scale of the plot. Indeed, on the high survival time side of the plot, the probabilities are very low and the distribution is very flat. The red zone of these plots corresponds to the area with at least 95 % the value of the maximum probability and the blue one is 85 % of the maximum probability.

On the left hand side of the plots, the distribution is highly multi-modal, with 2 areas of high probability, whereas on the right hand side, the spread of the high-probability area is wide, so the error bars are quite wide.

So we can conclude that the point sampling from the MDN offers no significant advantage over the standard neural network approach because of this multi-modal probability density.

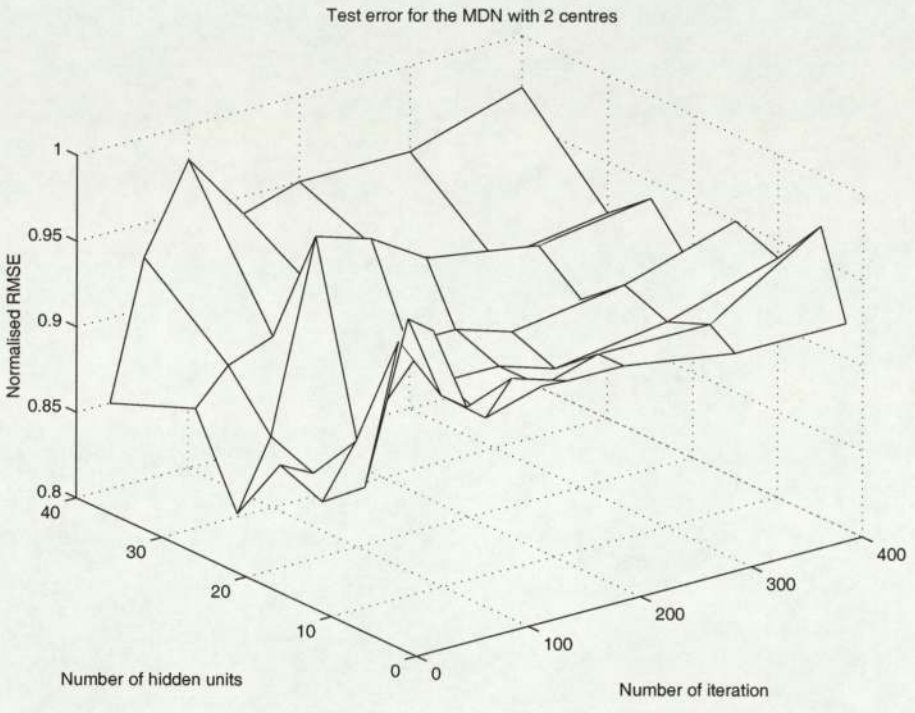

Figure 4.15: Test error for the MDN with 2 centers, the minimum test error is 0.82 reached with 25 hidden units and 30 iterations

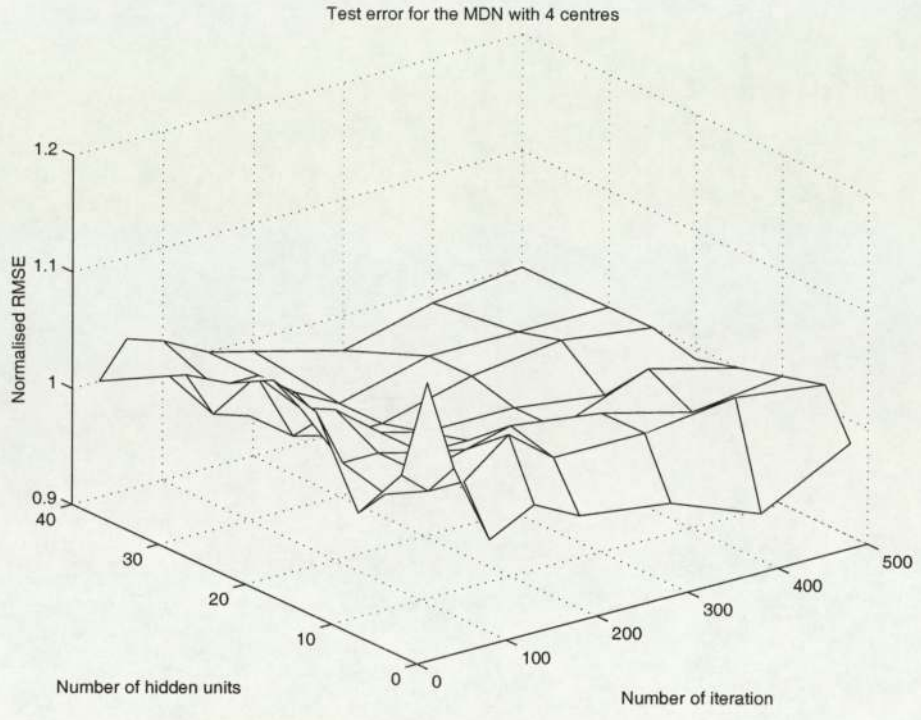

Figure 4.16: Test error for the MDN with 4 centers, the minimum test error is 0.94 reached with 2 hidden units and 400 iterations

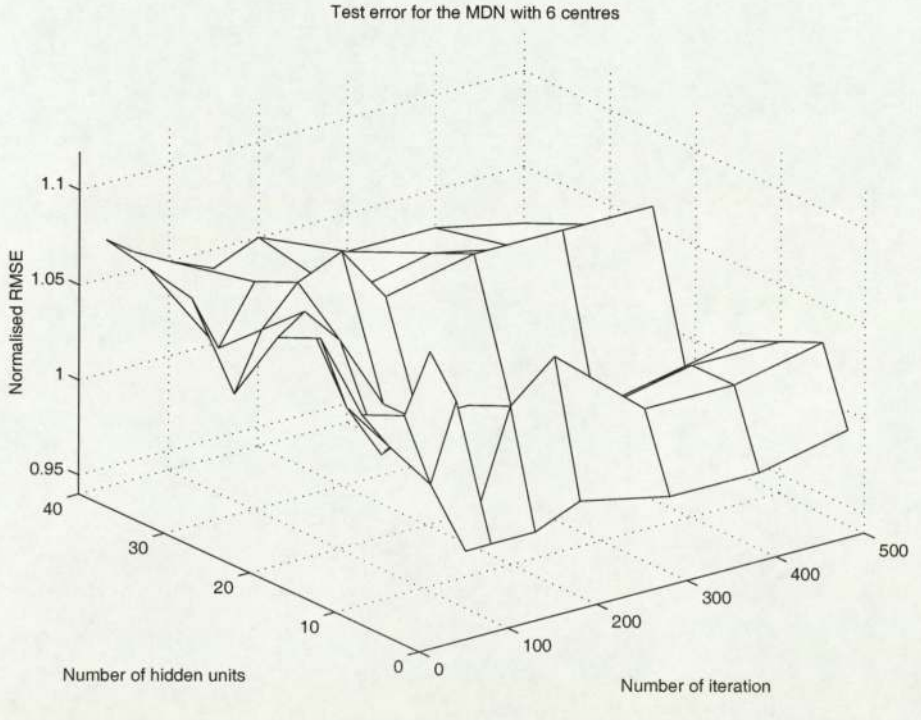

Figure 4.17: Test error for the MDN with 6 centers, the minimum test error is 0.95 reached with 20 hidden units and 400 iterations

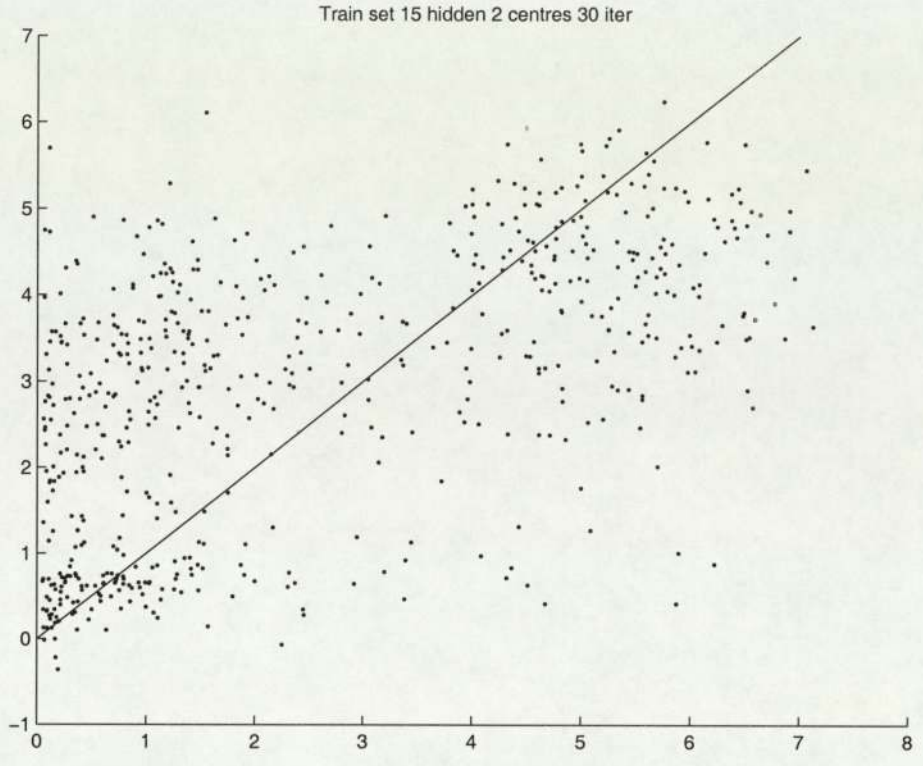

Figure 4.18: Plot of the target versus output for the MDN for the train set

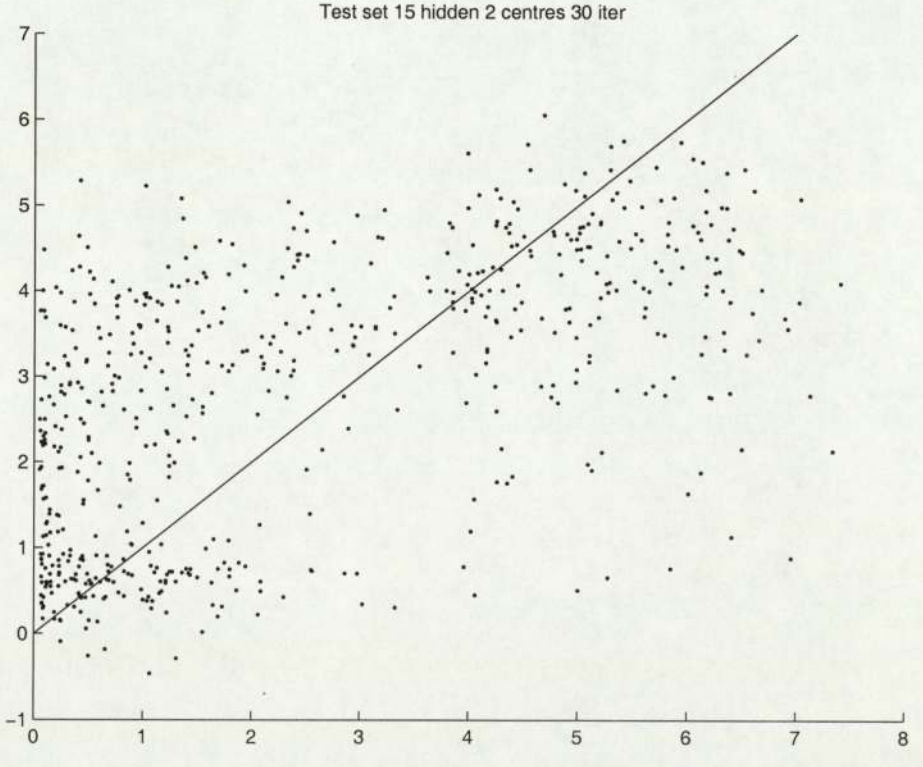

Figure 4.19: Plot of the target versus output for the MDN for the test set

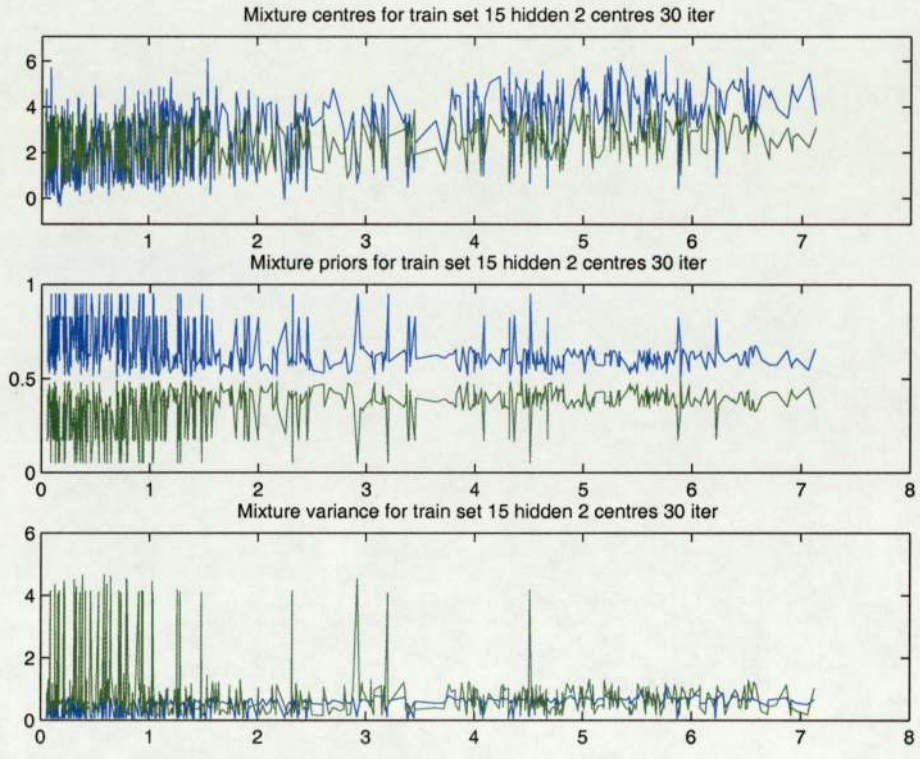

Figure 4.20: Plot of the centers of each Gaussian kernel, its prior and variance given the actual survival time for the train set

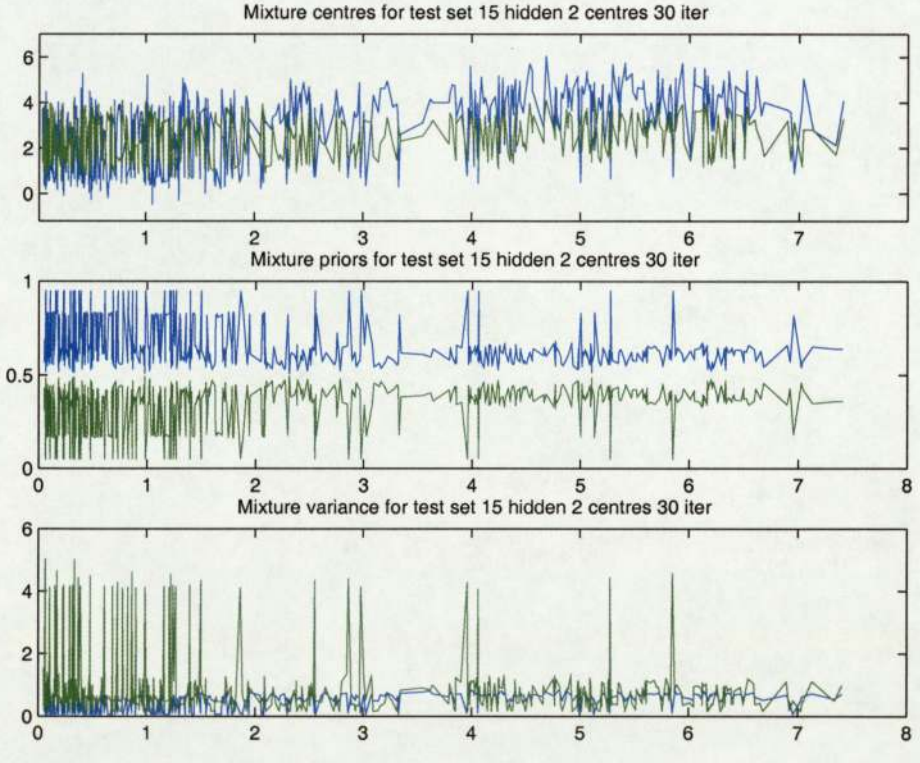

Figure 4.21: Plot of the centers of each Gaussian kernel, its prior and variance given the actual survival time for the test set

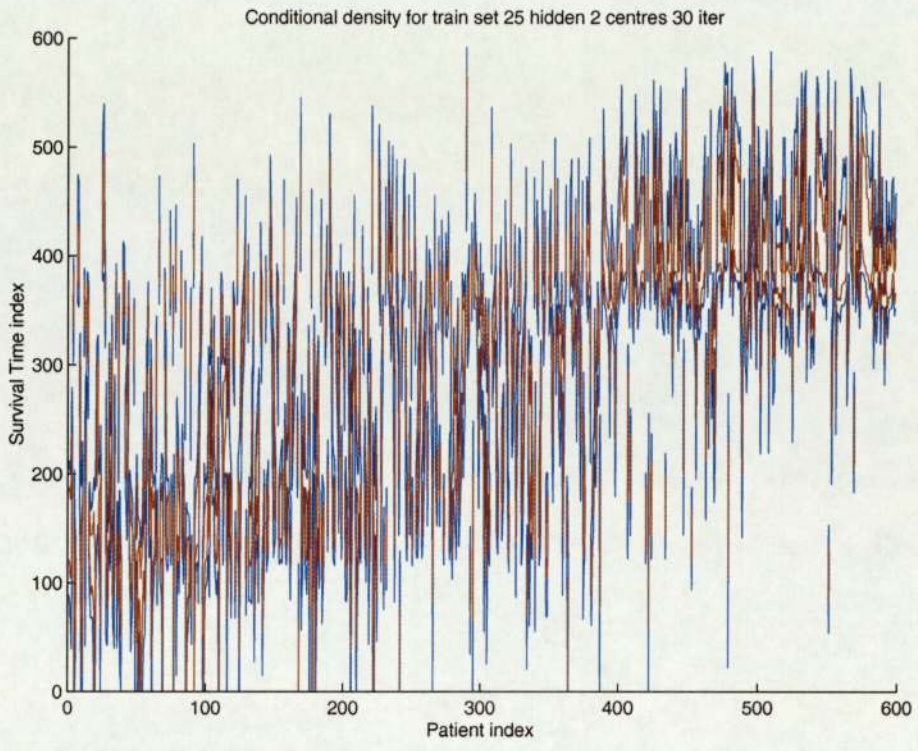

Figure 4.22: Plot of the conditional probability of the survival time for each patient. The patients are ordered by Survival Time for the train set

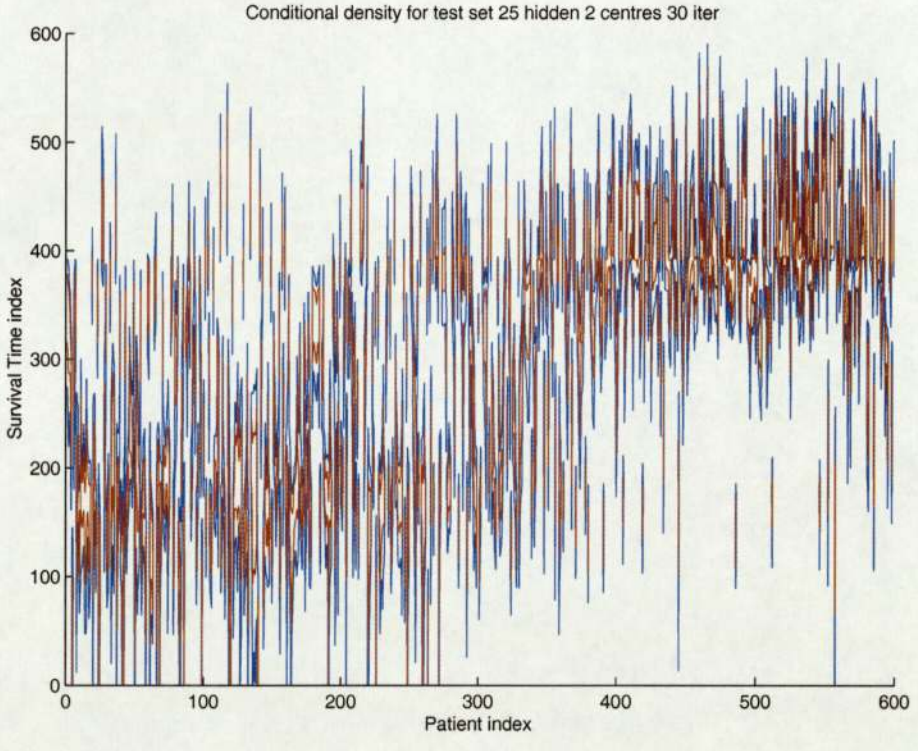

Figure 4.23: Plot of the conditional probability of the survival time for each patient. The patients are ordered by Survival Time for the test set

Exponential kernels

Since the Gaussian kernels provided only poor results, as for the MLPs, the use of exponential kernels was considered. Through a MDN, this model could offer good results by not being penalised by the outliers nor the long tail, and also the use of the right-sided approach introduced before could improve the model.

The exponential kernel function is defined as :

$$f(x) = \frac{\beta * \lambda}{\beta + 1} \exp(-\lambda funcexp(x)) \tag{4.31}$$

where λ is the age specific failure rate. This function integrates to 1 and has a mean of $\frac{\beta-1}{\beta\lambda}$. The plot of this function can be found on Figure 4.24. As β increases, the probability decreases more sharply and quickly for the negative values of x, i.e when the Survival Time is underestimated, the penalty increases with β.

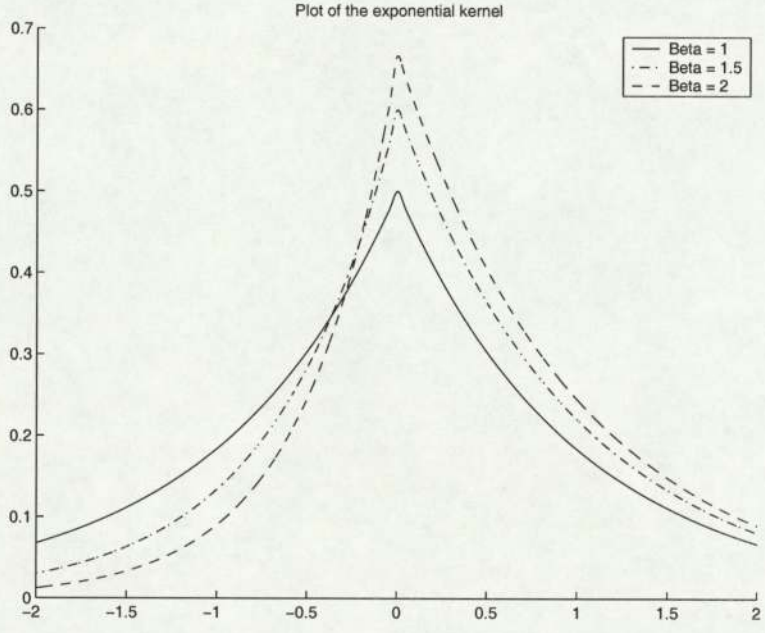

Figure 4.24: Plot of the exponential kernel with $\lambda = 1$ and different values of β

The kernels are defined as :

$$\phi_i(t|x) = \left(\frac{\beta * \lambda_i(x)}{\beta + 1}\right)^c \exp(-\lambda_i(x) * funcexp(t - \mu_i(x))). \tag{4.32}$$

The λ_i parameter is always positive, so:

$$\lambda_i = exp(z_i^\lambda) \tag{4.33}$$

The derivative of the negative log-likelihood is now

$$\frac{\partial E^q}{\partial \lambda_i} = -\pi_i * (\frac{c}{\lambda_i} - \lambda_i funcexp(t - \mu_i)) \tag{4.34}$$

And as :

$$\frac{\partial \lambda_i}{\partial z_i^\lambda} = \lambda_i \tag{4.35}$$

it comes

$$\frac{\partial E^q}{\partial z_i^\lambda} = -\pi_i \left(2c - 2\lambda_i^2 funcexp(t - \mu_i)\right) \tag{4.36}$$

Finally to conclude the computation of the derivative :

$$\frac{\partial E^q}{\partial z_{ik}^\mu} = \pi_i \lambda_i^2 \frac{\partial funcexp(t_k - \mu_{ik})}{\partial \mu_{ik}} \tag{4.37}$$

The optimum test error with these kernels is reached with 2 centers again, and β equals to 1.25 as shown on Table 4.3. So with a MDN, the right-side model offers better results than the standard one, opposite to the MLP with an exponential cost-function.

	Value of β							
Number of centers	1	1.1	1.25	1.5	1.75	2	2.5	3
2	0.87	0.86	0.82	0.85	0.85	0.87	0.90	0.92
4	0.91	0.91	0.92	0.93	0.90	0.89	0.95	0.96
6	0.92	0.95	0.92	0.96	0.89	0.93	0.96	0.96
8	0.88	0.89	0.92	0.90	0.95	0.94	0.90	0.88

Table 4.3: Minimum values of the test set error for the MDN with exponential cost-function given β

The results with the best combination of number of hidden units, number of iterations and number of kernels are shown in Figures 4.25 and 4.26. The results are similar to the results previously obtained, but a sort of clustering appeared, with two main groups along the diagonal and two small clusters misclassified. In this case, all the kernels are used (see Figures 4.27 and 4.28):

- The first kernel with a low parameter is used to predict the first half of the patients
- The second with high peaks in its parameters is used for the high actual survival time.

The plots of the conditional probabilities 4.29 and 4.30 show once again a strongly multi-modal distribution, especially for the test set where the trend is not along the diagonal at all, whereas for the train set a global trend can be observed along the diagonal. This feature is quite surprising since the normalised test error is at its minimum with this amount of iterations, hidden units and kernels. Nevertheless there is overfitting as the train set has some of its features perfectly fitted by the network since a diagonal line appears in the plot, and so this Network with this kind of noise model seems unable to provide good generalization.

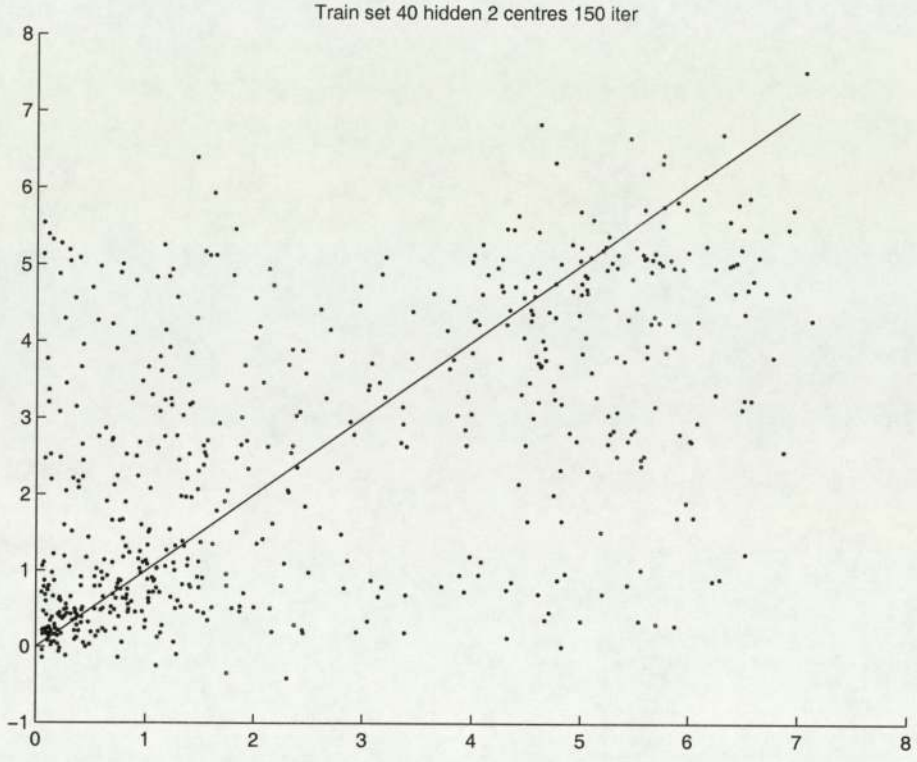

Figure 4.25: Plot of target versus output for the MDN with exponential kernel functions for the train set

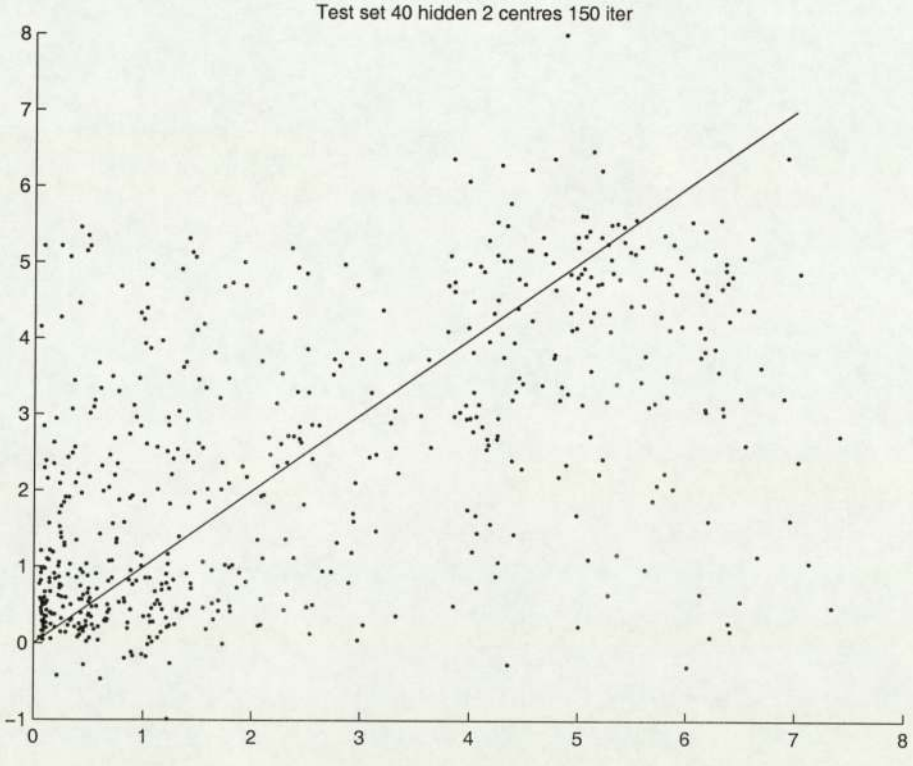

Figure 4.26: Plot of target versus output for the MDN with exponential kernel functions for the test set

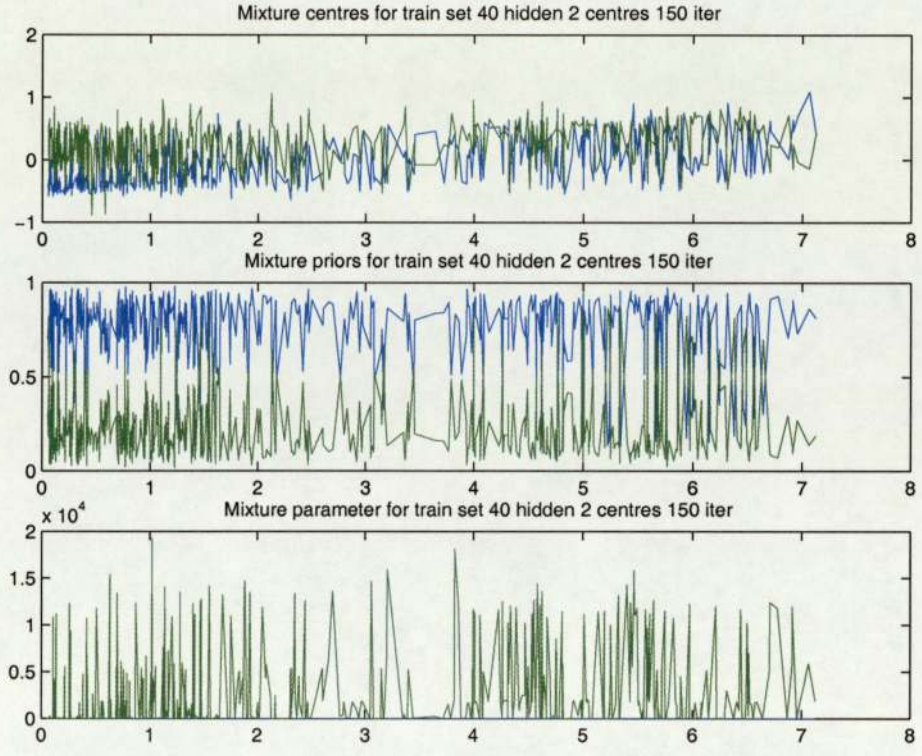

Figure 4.27: Plot of the center of each exponential kernel, their prior and parameter given the actual survival time for the train set

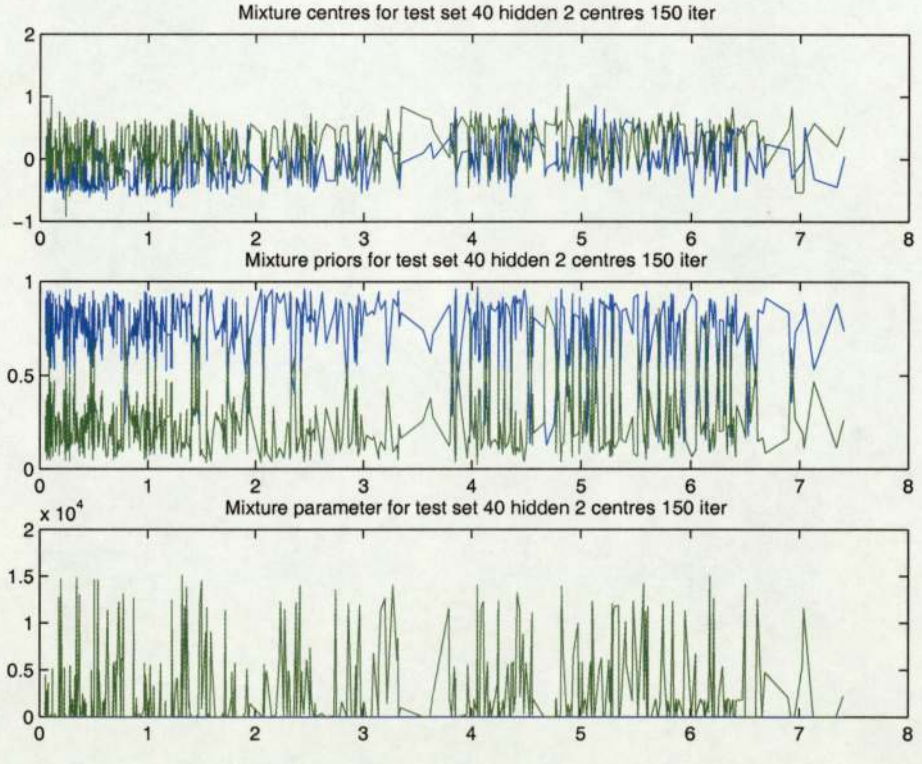

Figure 4.28: Plot of the center of each exponential kernel, their prior and parameter given the actual survival time for the test set

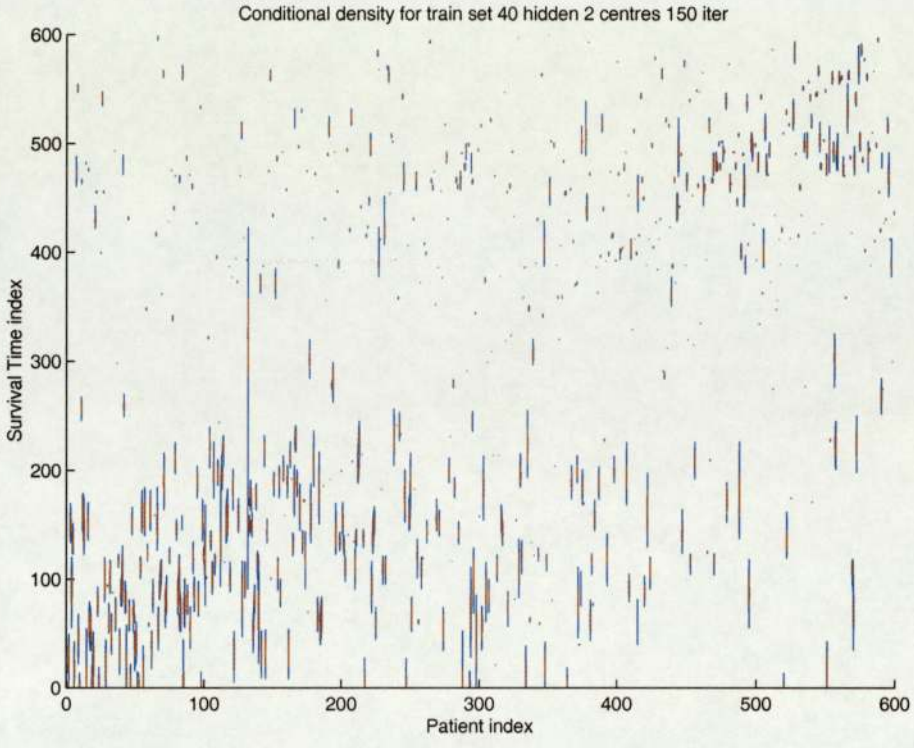

Figure 4.29: Plot of the conditional probability of the survival time for each patient. The patients are ordered by Survival Time for the train set

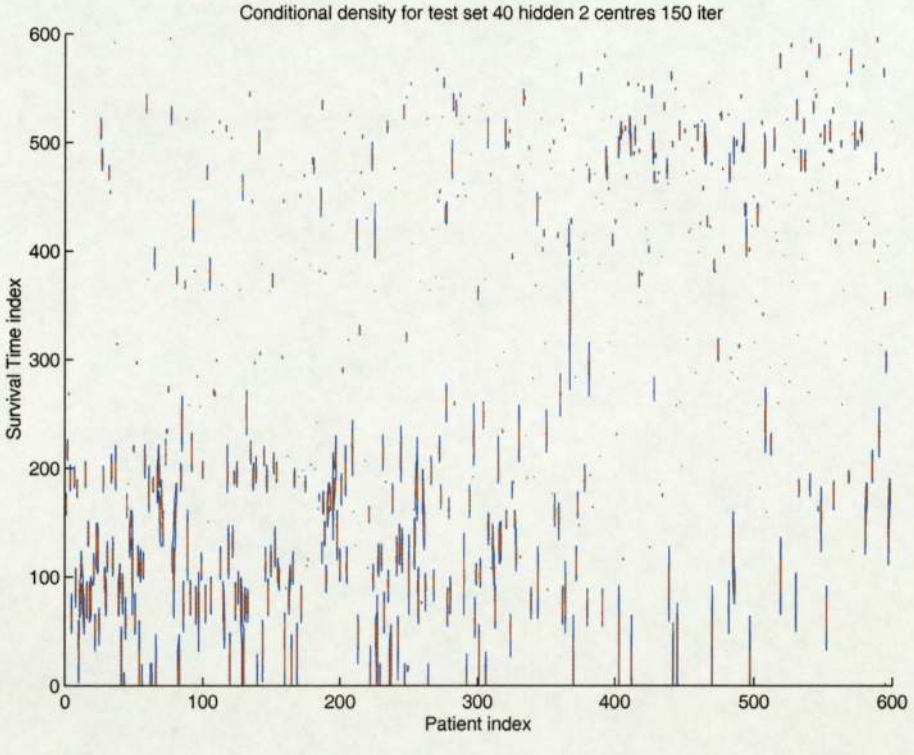

Figure 4.30: Plot of the conditional probability of the survival time for each patient. The patients are ordered by Survival Time for the test set

Erlangian distribution kernel

Another possible approach is to use the results of renewal theory[5]. Renewal theory is the study of probability problems connected with the failure and replacement of components, and can be extended to medical statistics where the failure could be the relapse or the death. The main goal is to be able to predict the *failure-time*, the time when the event failure occurs, which is a random variable T.

T is non negative, and its probability density functions (p.d.f.) is defined as

$$f(x) = \lim_{\delta x \to 0^+} \frac{prob(x < X \leq x + \delta x)}{\delta x} \tag{4.38}$$

with

$$\int_0^{+\infty} f(x)dx = 1. \tag{4.39}$$

One of the most used distributions in renewal theory is the Erlangian distribution, which will be used in this part. The p.d.f of this distribution is

$$\frac{\rho^\alpha (x-\mu)^{\alpha-1} exp(-\rho(x-\mu))}{\Gamma(\alpha)} \tag{4.40}$$

where μ is the origin of the function as it is defined only on $[\mu; +\infty[$ and ρ its parameter. The α parameter is the form parameter of the distribution. In the case $\alpha = 2$ which will be explored later, the function has two interesting features as shown on Figure 4.31

- A broad peak corresponding to the highest probability of dying,
- A long tail so that the probability of dying does not go to zero too quickly.

This function might fit the noise on the Survival Time, as we always know the minimum of Survival Time and the most likely true length of Survival is therefore greater than the one in the dataset.

In this case, the kernel functions are :

$$\phi_i(t|x) = \frac{\rho^\alpha (x-\mu)^{\alpha-1} exp(-\rho(x-\mu))}{\Gamma(\alpha)} \tag{4.41}$$

The parameter ρ as to be always positive, so it is considered as an exponential

$$\rho_i = exp(z_i^\rho) \tag{4.42}$$

The derivative of the log-likelihood is

$$\frac{\partial E^q}{\partial \rho_i} = -\pi_i \left\{ \frac{\alpha}{\rho_i} - max(0, (t-\mu_i)) \right\} \tag{4.43}$$

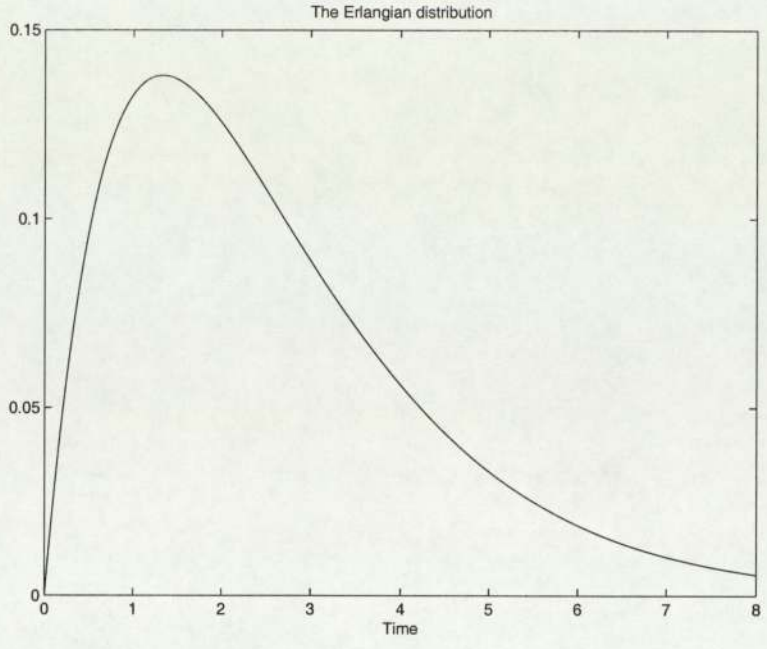

Figure 4.31: The Erlangian function with $\rho = 0.75$ and $\alpha = 2$

And as

$$\frac{\partial \rho_i}{\partial z_i^\rho} = \rho_i \tag{4.44}$$

we have

$$\frac{\partial E^q}{\partial z_i^\rho} = -\pi_i \left\{ \alpha - \rho_i max(0, (t - \mu_i)) \right\} \tag{4.45}$$

And finally

$$\frac{\partial E^q}{\partial z_{ik}^\mu} = \pi_i \left\{ \frac{\alpha - 1}{max(0, (t_k - \mu_{ik}))} + \rho_i \right\} \tag{4.46}$$

The initialisation of this MDN is made by setting all the priors with equal values, summing to one, by all the centers having the value 0 and by setting the ρ parameters to obtain the modes evenly distributed in the space of the target (see Figure 4.32). This allows for a large overlap of the kernels and simplifies the optimisation.

Here the actual prognosis is obtained by taking the mode of the Erlangian distribution with the highest prior. Taking the mode of the whole distribution would be useless as the distribution can have really high peak for high values of ρ as shown on Figure 4.32 : the function on the left of this plot has the highest value of ρ and its mode is much higher than the modes of the others. So the whole distribution is dominated by the function with the higher ρ, whatever its associated prior is. That is why there is no plot of the conditional density, as they are all strongly dominated by the low survival time.

The plots of target versus output 4.37 and 4.38 show four bars, corresponding to the four kernels as the functions are defined only from their origin and have high peak,

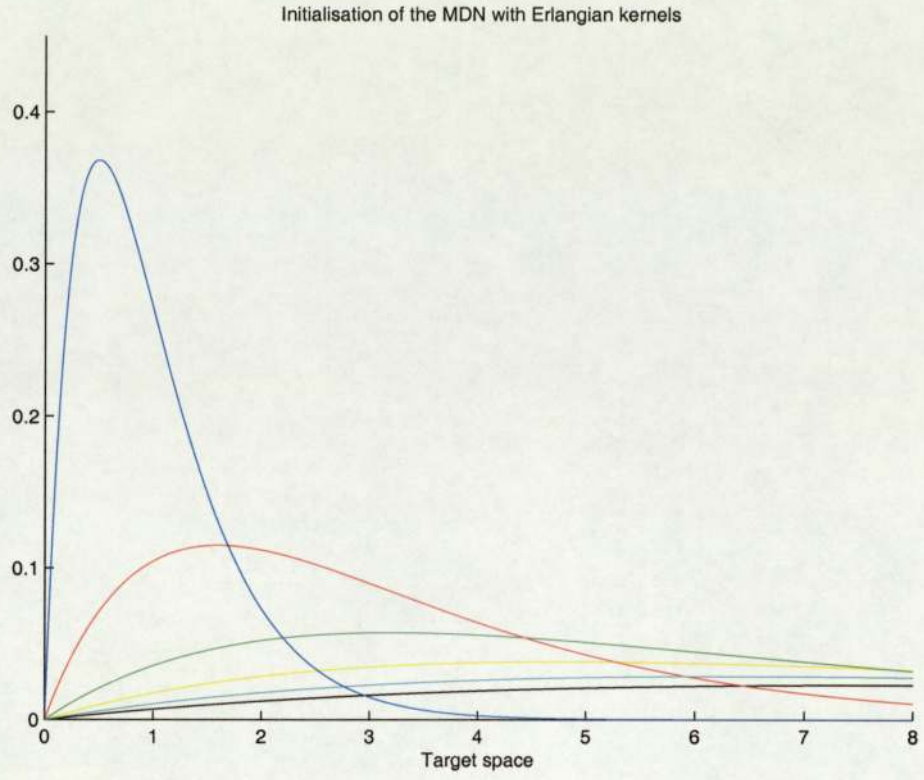

Figure 4.32: Initialization of the Erlangian distribution

especially for the low survival time, when ρ is high. Two main clusters can be seen on these scatter plots :

- One at the low actual survival time, where a lot of points are grouped under 1 year predicted
- One at the high actual survival time, where a lot of points are spread in the band 3-5 years predicted.

Very few are really misclassified, but this model misses the global trend of survival to provide only clusters, as a classifier would.

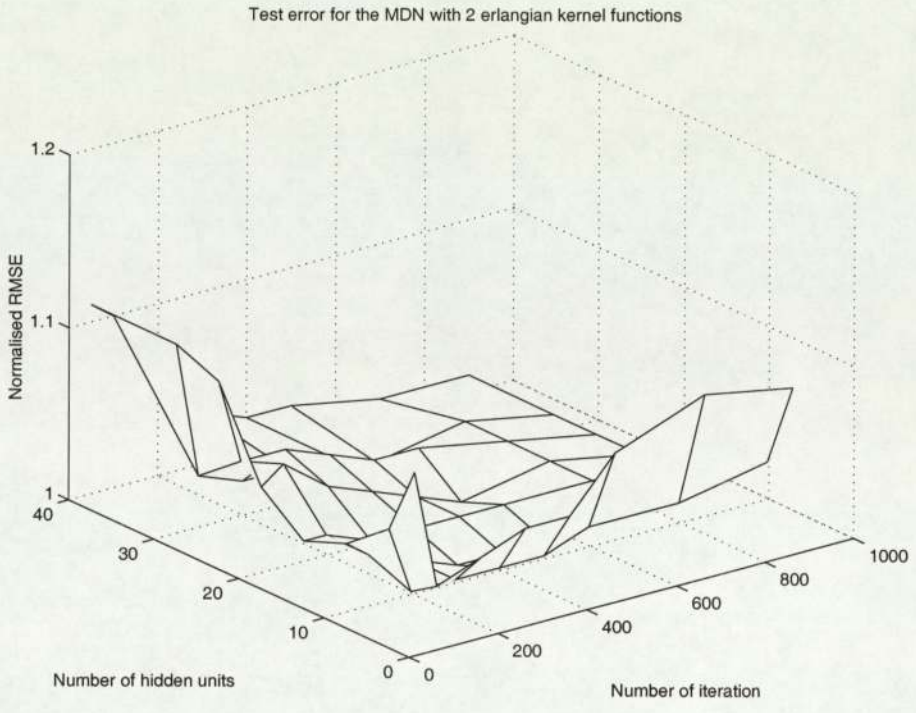

Figure 4.33: Test error for the MDN with 2 erlangian kernel functions, the minimum test error is 1.01 reached with 10 hidden units and 300 iterations

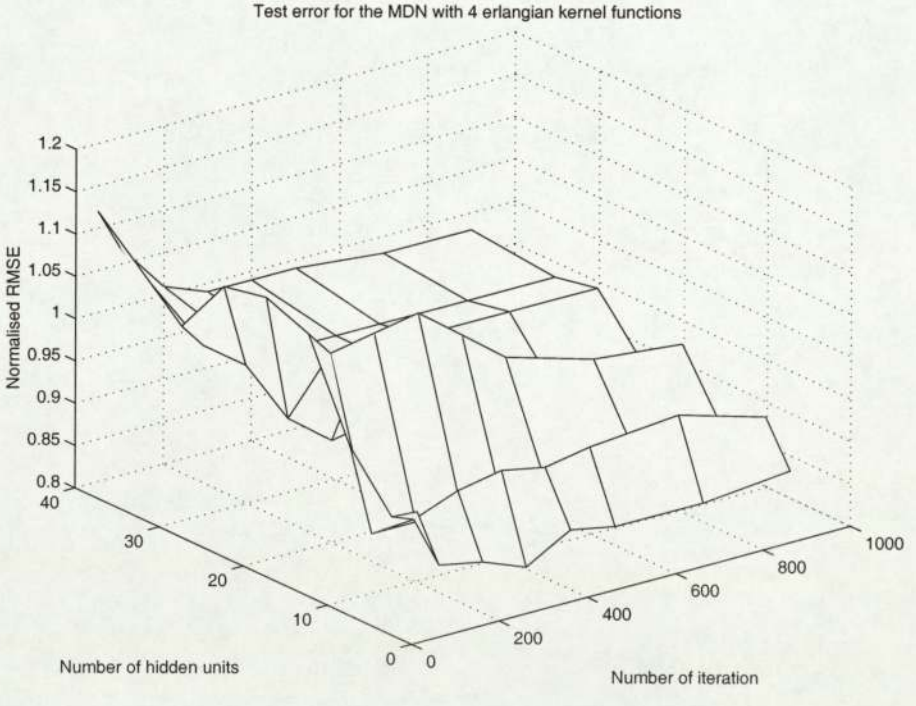

Figure 4.34: Test error for the MDN with 4 erlangian kernel functions, the minimum test error is 0.84 reached with 2 hidden units and 300 iterations

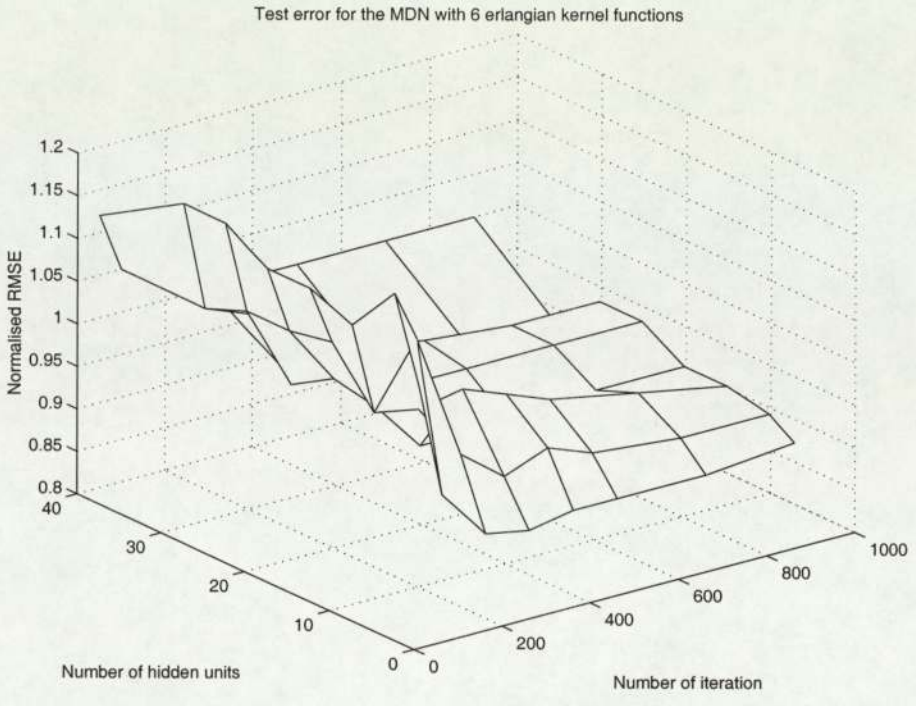

Figure 4.35: Test error for the MDN with 6 erlangian kernel functions, the minimum test error is 0.89 reached with 2 hidden units and 300 iterations

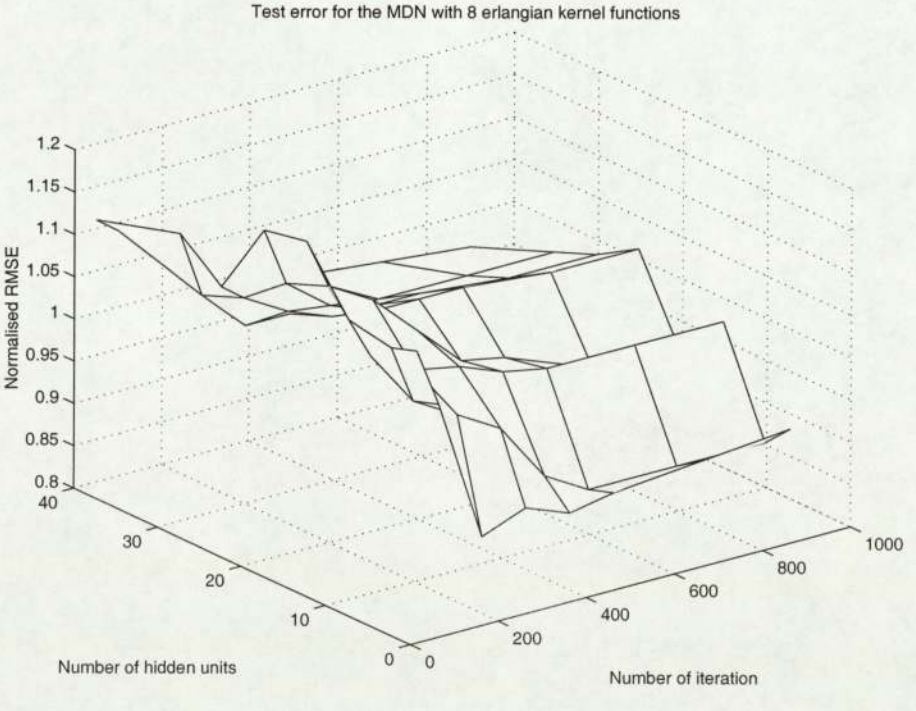

Figure 4.36: Test error for the MDN with 8 erlangian kernel functions, the minimum test error is 0.89 reached with 2 hidden units and 200 iterations

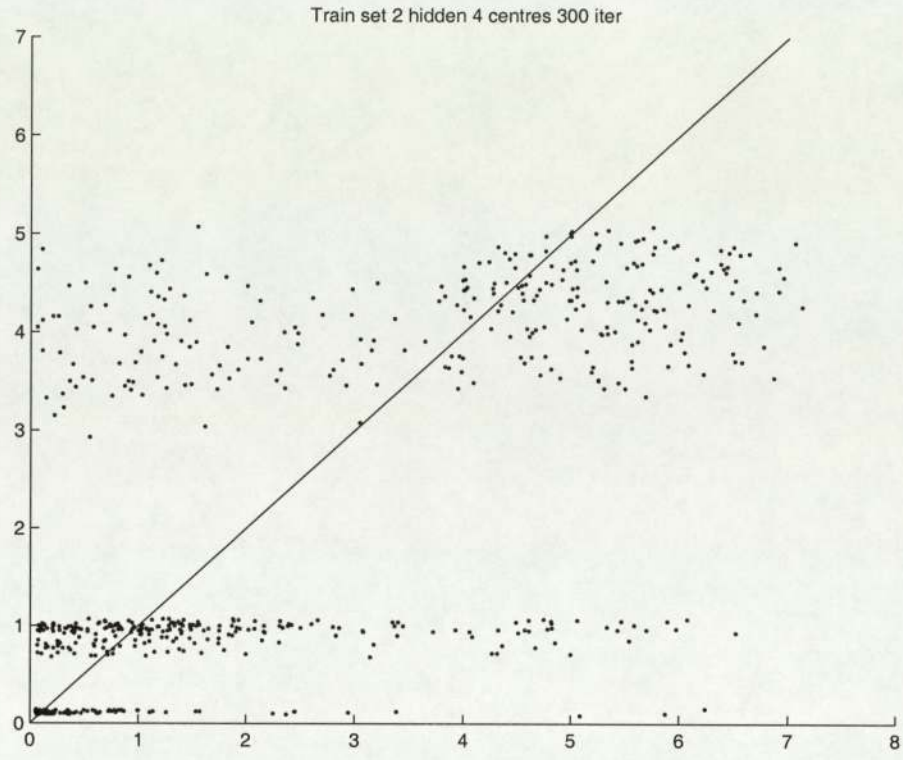

Figure 4.37: Plot of target versus output for the MDN with Erlangian kernel functions for the train set

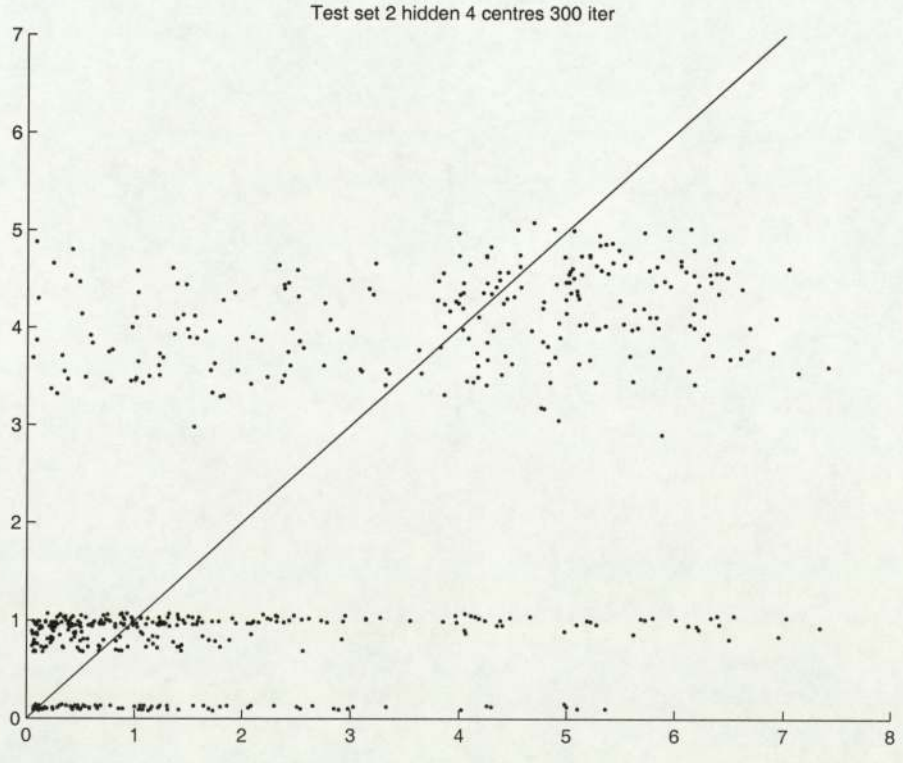

Figure 4.38: Plot of target versus output for the MDN with Erlangian kernel functions for the test set

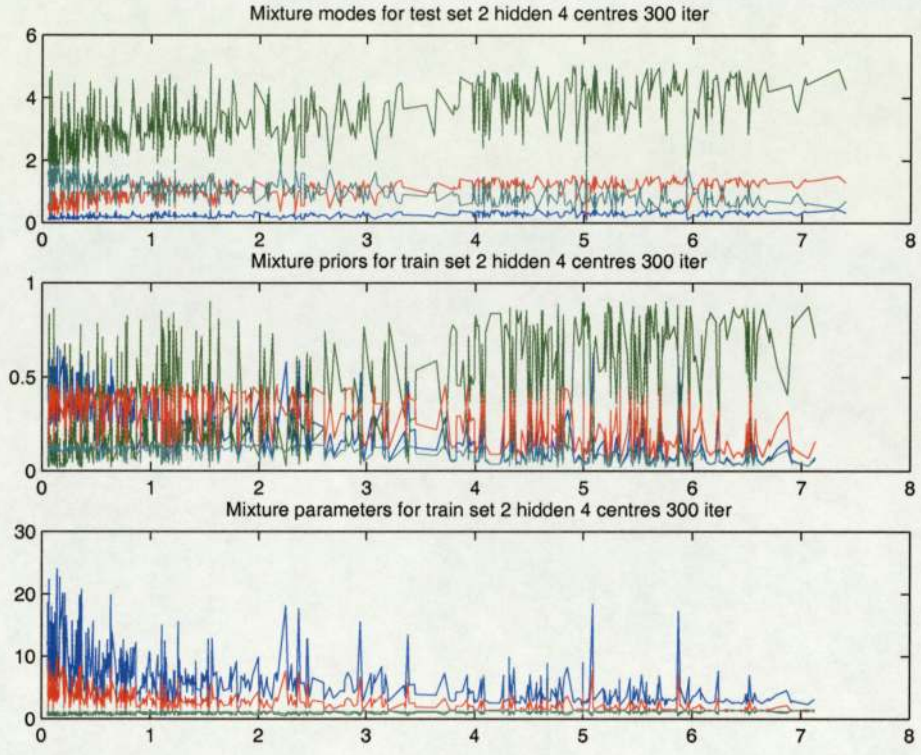

Figure 4.39: Plot of the mode of each Erlangian kernel, their prior and parameter given the actual survival time for the train set

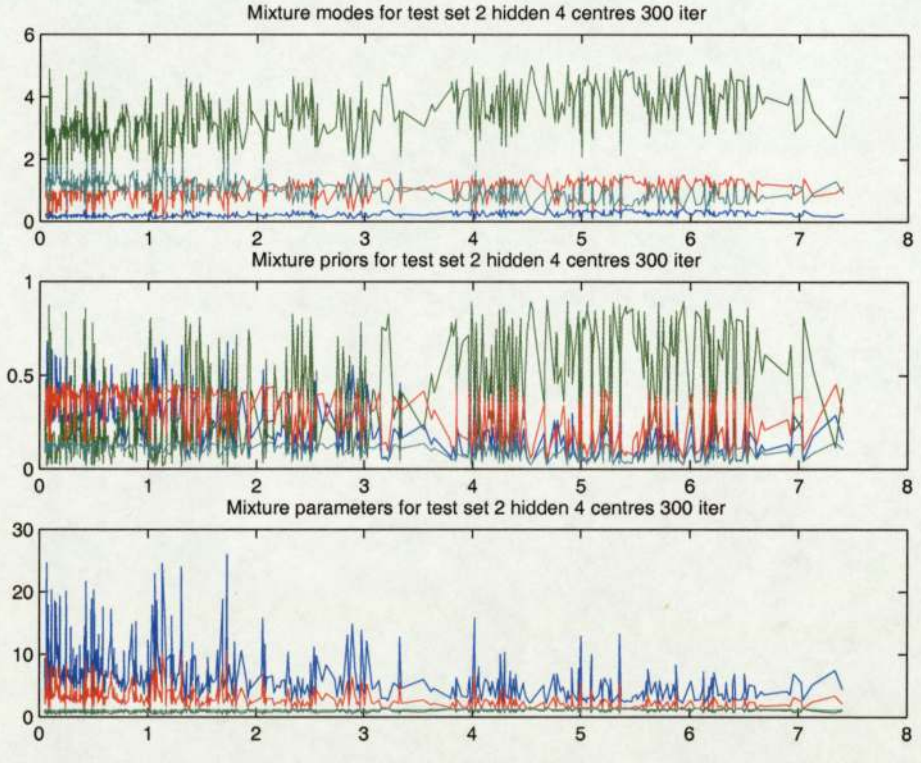

Figure 4.40: Plot of the mode of each Erlangian kernel, their prior and parameter given the actual survival time for the test set

4.3 The classification approach

So far, the results have been disappointing. So rather than trying to estimate the Survival Time, trying to separate the patients into different clusters was considered. To perform this, the data set was divided in c different classes corresponding to the different features we want to extract. The most interesting features would be, in our case, to predict if a patient will live longer than a fixed period, or in which period of Survival Time she is most likely to be. The results with classification are likely to be better than with the regression because the cluster size is bigger, and the output is less sensitive to the noise.

4.3.1 The error measure

When using a classifier, the straightforward error measure is the misclassification rate, i.e. the number of patients which are classified as being in one class while belonging to another

$$Err = \frac{1}{n}\sum_{q=1}^{n} ||t^q - y^q||^2 \tag{4.47}$$

where n is the number of patterns, t is the target and y is the network output.

4.3.2 The Neural Networks

The most standard classifier is the standard MLP with c binary outputs, corresponding to each of the classes. The likelihood of the observations is :

$$L = \prod_{k=1}^{c}\prod_{q} p(x_k^q, t_k^q) = \prod_{k=1}^{c}\prod_{q} p(t_k^q|x_k^q)p(x_k^q) \tag{4.48}$$

or

$$L = \prod_{k=1}^{c}\prod_{q} (y_k^q)^{t_k^q} p(x_k^q) \tag{4.49}$$

Finally, the target values for a 1-of-c coding scheme are binary, so are far from having a Gaussian distribution.

Once again, the probability of the input is independent of the parameters of the models, so we have

$$E = -\sum_{k=1}^{c}\sum_{q} t_k^n \ln(y_k^q) \tag{4.50}$$

The outputs of the MLP are considered as probabilities to belong to one class, so they should sum to unity and lie between 0 and 1. Indeed, using Bayes' rule, the posterior probability of class C_k is :

$$p(C_k|z) = \frac{p(z|C_k)p(C_k)}{\sum_j p(z|C_j)P(C_j)} \tag{4.51}$$

where z is the activation vector of the last layer of units.

This can be enforced by using a *softmax* activation function

$$y_k = \frac{\exp(a'_k)}{\sum_j \exp(a'_j)} \tag{4.52}$$

where a'_k is defined as

$$a'_k = a_k + \ln P(C_k) \tag{4.53}$$

and a_k is the summed inputs of the k^{th} output unit, or the output of the network in the case of linear output hence corresponding to $p(z|C_k)$[2].

So using this activation function, the prior probability of belonging to a class is included in the cost-function and prevent one of the most annoying features of the classifier which is to have a bias toward the biggest class[7].

4.3.3 Classification in time

The first attempt with the classifier was to try to predict in which period of Survival Time a patient is most likely to be. The Survival Time has been divided into c classes, containing the same number of patients but with a different time span. Indeed the Figures 4.41, 4.44 and 4.47 show the length of each of the classes (vertical line) altogether with the censored patients (dots on line with 1 on the vertical axis) and the patients with deadly outcome (0 on the vertical axis).

With only two classes, the results are quite good as the classifiers can predict if the patient is to survive at least 1.7 years with only 22% of error (Figure 4.42). Moreover, the confusion matrices in Figure 4.43 show small confusion, and there is no bias toward any of the two classes since the number of misclassified are about the same for each class.

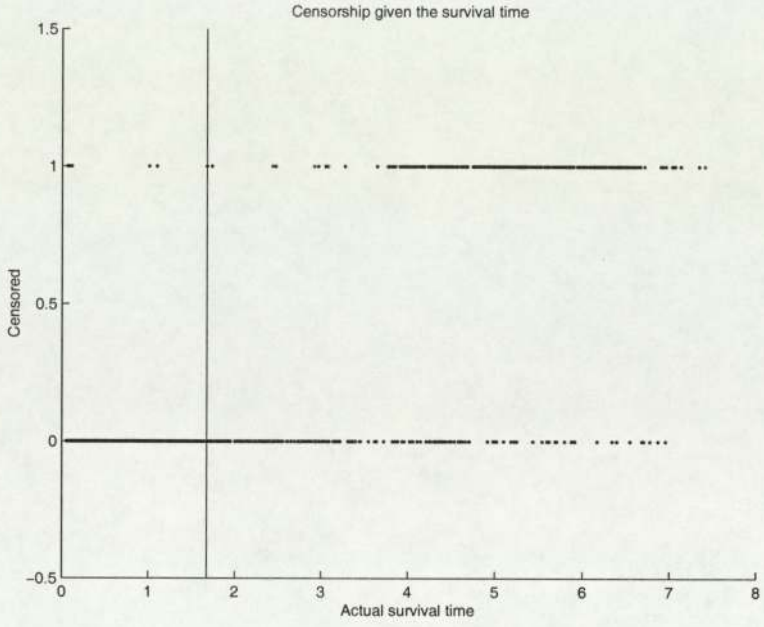

Figure 4.41: Censored patients given the actual survival time

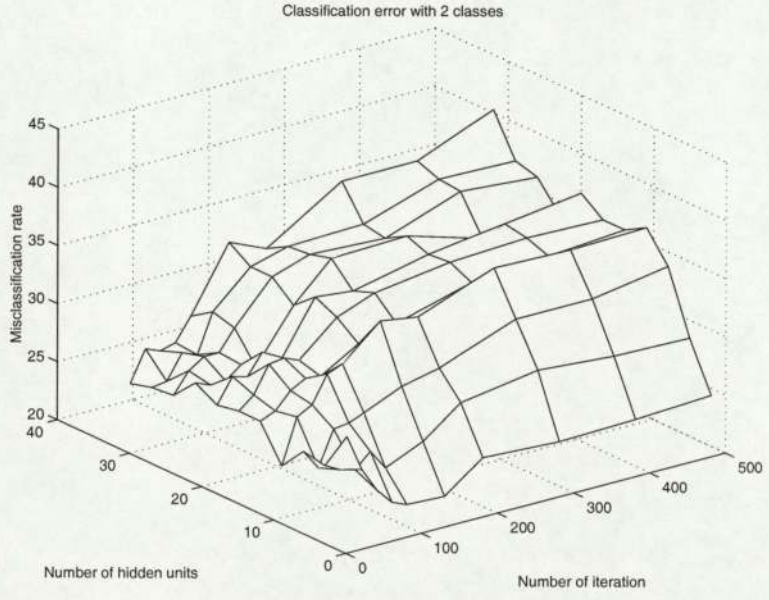

Figure 4.42: Test error for the classifiers with 2 classes of equal size, the misclassification rate on the test set is 22% reached with 2 hidden units and 150 iterations

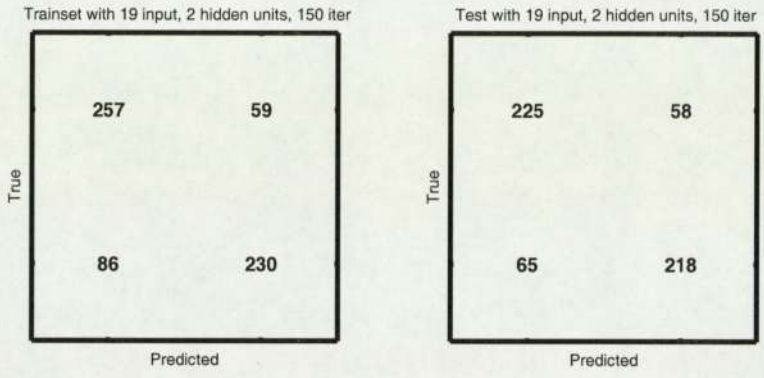

Figure 4.43: Plots of the confusion matrices on the train and test set with 2 classes

With three classes, the results are a bit worse. As shown in the confusion matrices on Figure 4.46, the MLP cannot recognize the second class. Indeed, patients belonging to the second class are mainly (mis)classified in the first class. Less than a third of them are correctly classified, and another third is classified in the class corresponding to the longest survival time. Nevertheless, the first class and the last class are still properly recognized, resulting in an overall misclassification error of only 40% (Figure 4.45).

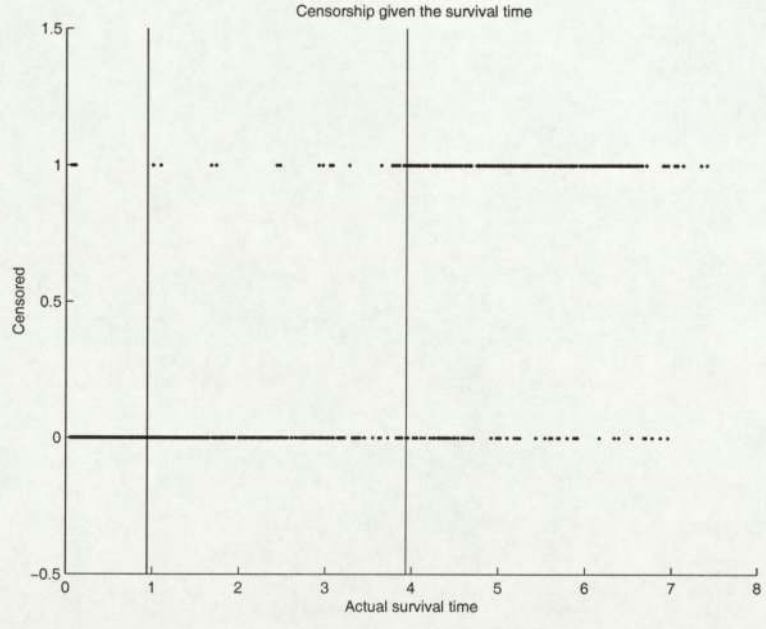

Figure 4.44: Censored patients given the actual survival time

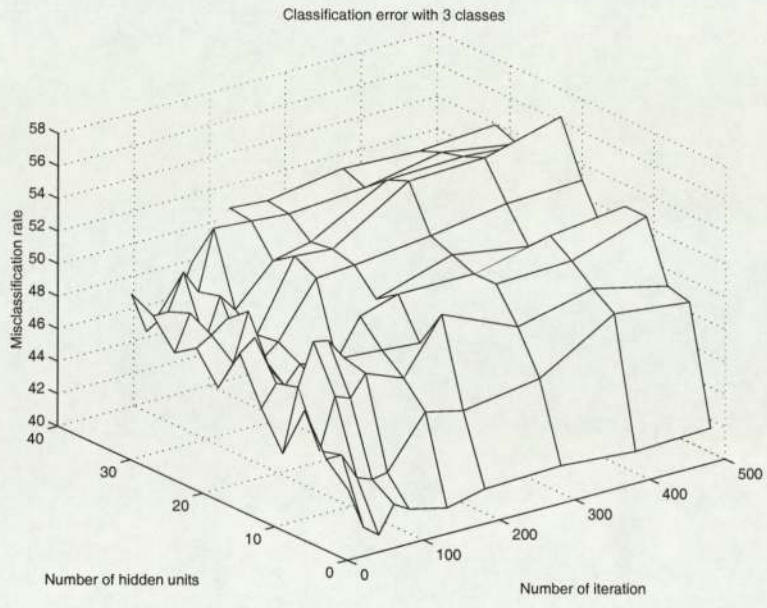

Figure 4.45: Test error for the classifiers with 3 classes of equal size, the misclassification rate on the test set is 40% reached with 2 hidden units and 60 iterations

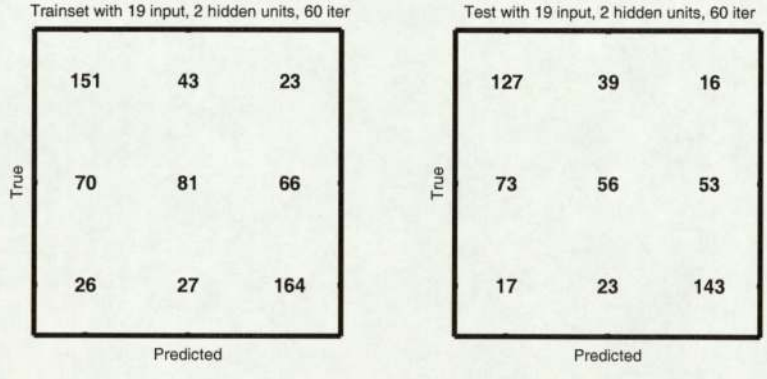

Figure 4.46: Plots of the confusion matrices on the train and test set with 3 classes

Again, with four classes, the results are quite disappointing as only the patients of the first class (lowest survival time, less than 8 months) and the last class (more than 4.5 years) are not mostly misclassified (Figure 4.49). The patients of the second class are classified one time over three in the right class, one over three in the first class, and the remaining evenly between the last two classes. Worse, the patients of the third class are nearly always misclassified, and half of the time they are classified as being in the last class. The overall test error is 54 % (Figure 4.48), so the classifier is misclassifying more than classifying properly.
One may believe the different time length of the classes is the reason for the misclassification, but the third class, the longest, is least recognized by the Neural Network. Moreover the Network classifies the patients of the second class as often in the first as in the second class, while the first class is shorter. So the time length of the classes is

not the issue.

Figure 4.47 shows the partition of censored patients and patients with deadly outcome given the actual survival time, altogether with the class length. This graph emphasizes the increasing number of censored patients with the actual survival time, and therefore with the increase of the class index. So maybe a classification with only the patients with deadly outcome would perform better as it would remove a large amount of noise on the target.

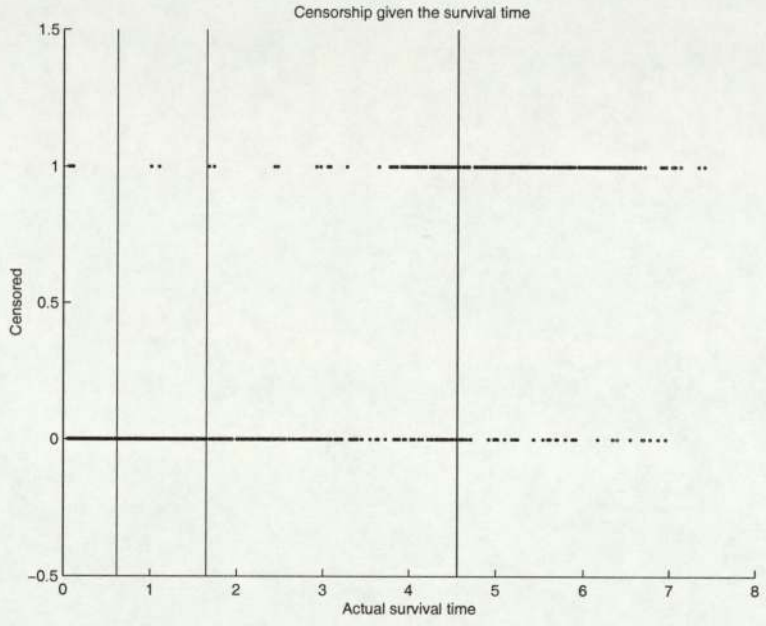

Figure 4.47: Censored patients given the actual survival time

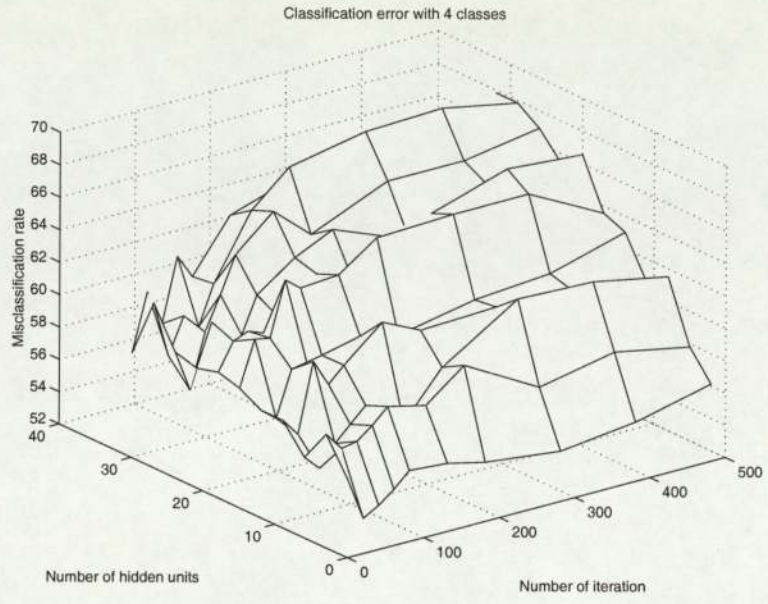

Figure 4.48: Test error for the classifiers with 4 classes of equal size, the misclassification rate on the test set is 54% reached with 2 hidden units and 60 iterations

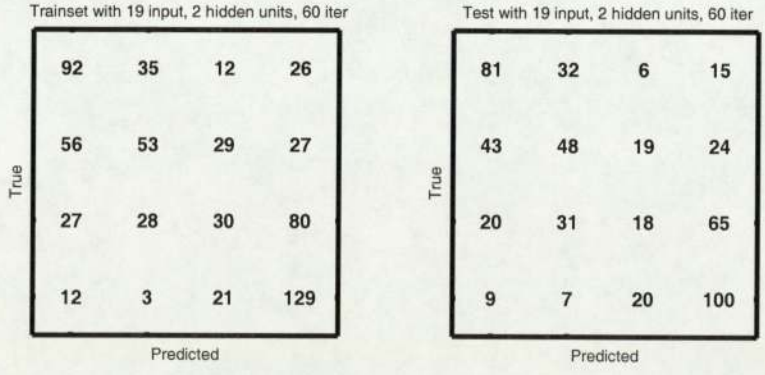

Figure 4.49: Plots of the confusion matrices on the train and test set with 4 classes

4.3.4 Classification in time with only the patients with deadly outcome

By removing the censored patients, the noise level on the output has been mainly set aside. Indeed in the last but one class in each previous case, a lot of patients are censored, so are more likely to live longer and be in the higher classes. It can be assumed that in this case the error rate would be lower.

Strangely, the results are not improved, and worse, the classification rate dramatically decreases, as shown on Figures 4.50 and 4.52. On the two classes problem, only two thirds of the patients are properly classified (Figure 4.51), whereas four fifths were correctly classified if the censored patients were kept. With three classes, only half of them are correctly classified (Figure 4.53), the other half being equally spread over the two others classes.

So it seems that the prediction of a Survival Time is impossible with this dataset.But what about the prediction of deadly outcome or censorship, which could be viewed as being cured.

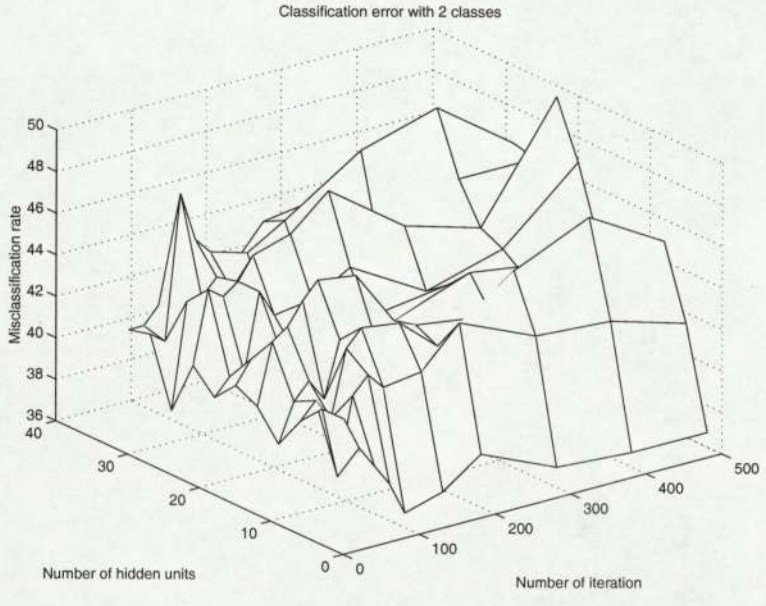

Figure 4.50: Test error on the classification between censored patients and patients with deadly outcome, the misclassification rate on the test set is 37% reached with 2 hidden units and 500 iterations

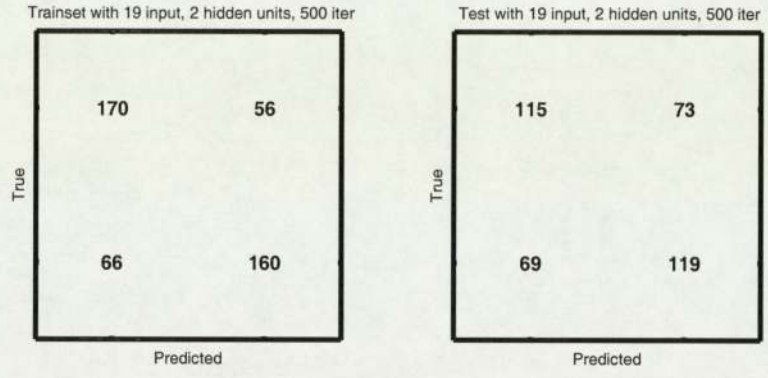

Figure 4.51: Plots of the confusion matrices on the train and test set with 2 classes

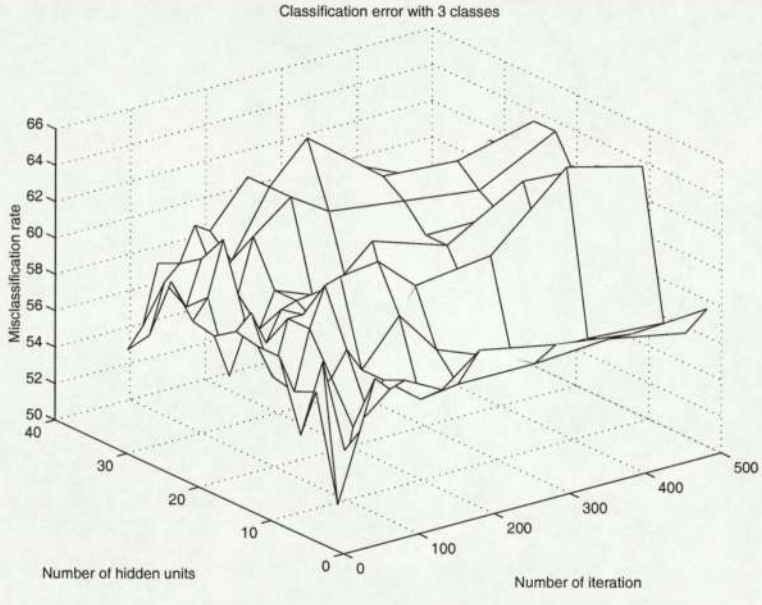

Figure 4.52: Test error on the classification between censored patients and patients with deadly outcome, the misclassification rate on the test set is 51% reached with 5 hidden units and 40 iterations

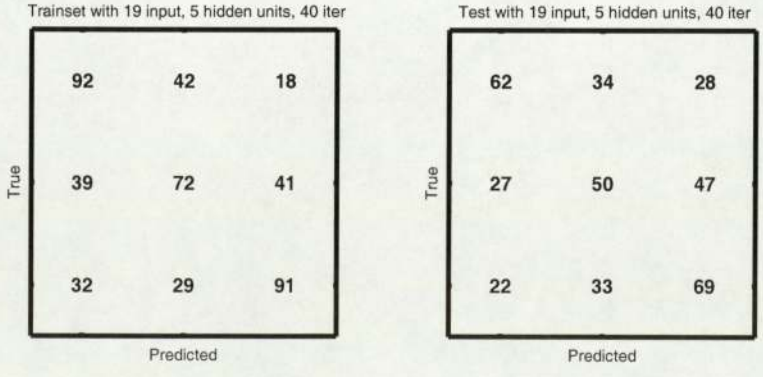

Figure 4.53: Plots of the confusion matrices on the train and test set with 3 classes

4.3.5 Classification censored patients vs patients with deadly outcome

The goal of this experiment is to predict whether a patient will be censored or not. So as the number of censored patients is different of the number of patients with a deadly outcome, the classes have no longer the same size.

Figure 4.55 shows that very few of the censored patients are misclassified, with only one sixth considered as having a deadly outcome by the network, and a fourth of the patients who did not survive are classified as being censored. So globally the results are quite good, with 20% overall misclassification rate (see Figure 4.54). On the other hand if someone is predicted to be censored, the confidence is this prediction is only two thirds whereas is someone is predicted to have a deadly outcome, the prediction is true 90 % of the time.

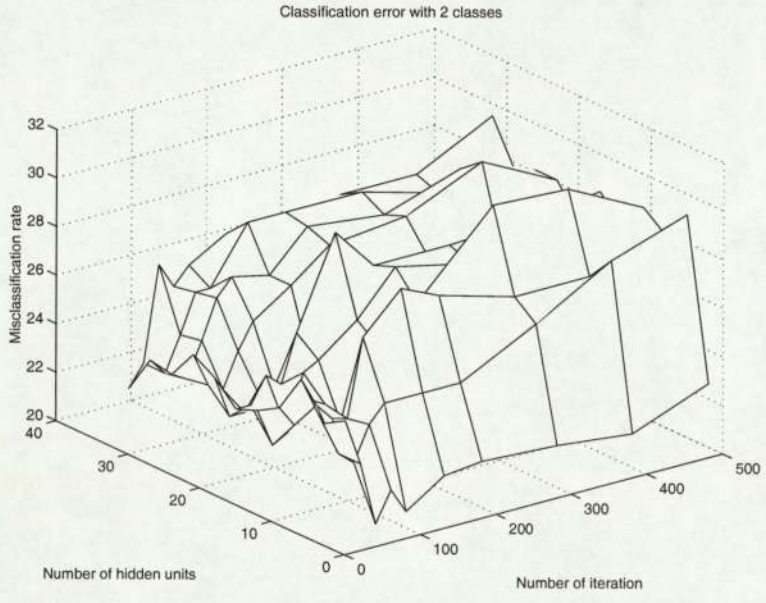

Figure 4.54: Test error on the classification between censored patients and patients with deadly outcome, the misclassification rate on the test set is 20% reached with 2 hidden units and 60 iterations

Unfortunately, this experiment is not relevant because :

- A censored patient cannot be confirmed to be a patient who survives.
- The spread of the censored patients is quite wide, most of them being after 4 years of actual Survival Time, but some lying between 2 months and 4 years.

Although this issue is interesting, due to lack of time, we could not explore it any further.

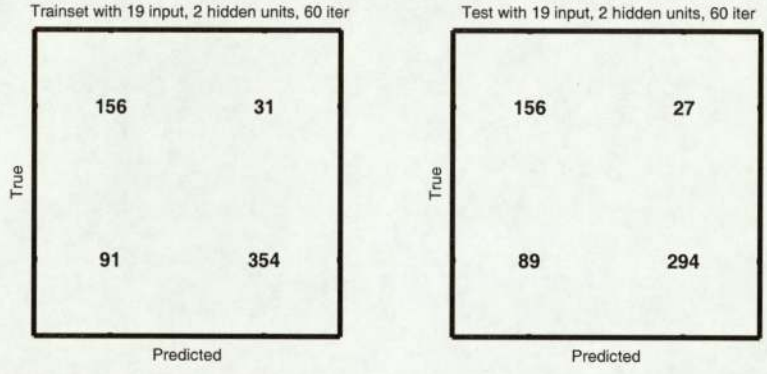

Figure 4.55: Plots of the confusion matrices on the train and test set for the classification between censored patients and patients with deadly outcome

ASTON UNIVERSITY
LIBRARY & INFORMATION SERVICES

Chapter 5

Discussion and Conclusion

In this thesis, we used several different Neural Network methods to try to predict the survival time of patients suffering from the Ovarian Cancer. This problem is not an easy one, obviously, as it is the case for most of the medical problems. First of all, the dataset contained some missing data and was filled for medical purpose, so it needed recoding and filling in of the missing values by estimating what would be these missing values.

Then several variables selection methods were performed to choose only the most relevant inputs. Thus it helps keeping low the complexity of the network. It also improves the training method. Furthermore, these methods allow the reduction of the needed amount of data to be collected on the patients, and ease the data extraction. Other methods, such as feature selection, could be used and may provide better results. The feature selection combines several variables into a feature and reduces the dimensionality of the inputs for the neural networks. Moreover, with this type of powerful preprocessing, expert knowledge can be used and could dramatically improve the results.

Once all the data has been prepared for the analysis, standard point estimators, the MLPs and RBF networks, which are typical tools for prediction, were used. However, they provided disappointing results, even when tuned to fit the characteristics of the dataset. Indeed the error measure remained consistently over 0.7.

Hence, another approach was considered using the full distribution of the output probabilities, thus providing much more information than the point estimators. The distributions of the Survival Time were obtained through MDNs which combine an MLP and a mixture model. Unfortunately, these distributions occur to be highly multi-modal and quite wide. This explains why the point estimators failed to predict reliably the Survival Time. However, a global trend can be outlined.

The classification attempts show consistent results with the regression case, but the strange discrimination of censored patients vs patients with deadly outcome could not be explained.

The output probabilities of the MDNs, although multi-modal, have a global trend along the true probabilities. So the MDN is able to learn some of the features of the dataset, but misses some hidden information since other modes appear. Hence, we can conclude that some information is lacking, for example the psychological state of a patient which can change the reaction to the treatment a lot and thus the outcome. Moreover, a lot of medical predictions using neural networks include more precise physiological data such as blood proteins measurements. Indeed, there are some blood screening tests for ovarian cancer involving 2 proteins, the CA 125 and OVXI, which are known to have a diagnostic potential[1].

This may partly explain why the results obtained are so disappointing, whereas other studies involving cancer and neural networks provide exciting results. For example the best model developed in [9] achieved an accuracy of around 79 %, but it involved a much more complex preprocessing by using a hazard model .

Nevertheless, the techniques used in this thesis can be considered as standard methods for further work with neural networks in medical problems as they provided interesting results.

The issues of safety and reliability have not been raised in this thesis since the predictions of the networks are not yet suitable for medical usage. Nevertheless, one should bear these issues in mind as the use of such predictions in a practical setting is not straightforward. Indeed, before being used, such methods need to be validated. Since predicting the outcome to a ailment is a really sensitive issue because the prediction impacts on both treatment on patients and their psychological state, the doctors need to be conviced with both extensive testings on new patients and with methods which have been shown to be efficient. The simplest methods are the ROC curve in the classification approach, which is also the most commonly used by medics, and the error bars in the regression approach.

Moreover, thanks to the MDN, the full conditional probability of the survival time can be plotted and thus medics can easily spot what is the most likely survival time, and if other likely survival times appear, they can use their expert knowledge to decide whether to trust the network or not in such case, or just use it as a complement to the doctors expertise in case of doubt.

[1] http://pathology2.jhu.edu/ovca/

Appendix A

Description of the dataset

The data was provided to us by Dr Judy Powell and Dr Sean Kehoe of the Birmingham Women's Hospital and involved patient records of 1426 patients over a 7-year period of study.

NAME	DESCRIPTION	CODES
ID	Arbitrary case identifier	
AGE	Age in years	
AGP	quinary ageband	1 to 20 1=0-4 years, 2=5-9 years etc.
DIST	residence: DHA subregion	e.g. 57CA
DHA	Residence: DHA	1 to 22 1-11, rural, 12-22 within WM county
ICDO-M	ICDO morphology code	8000-9110 e.g. 8450=papillary cystadenocarcinoma
ICDO-B	ICDO behaviour code (modified)	0 "malignant" - but no biopsy 1 borderline malignancy 3 malignant 4 malignant, mod/well differentiated 6 malignant, metastatic site 8 malignant,poorly differentiated
DAN	anniversary date (diagnosis date)	
DLAST	date last seen alive or date death	
STAT	vital status on dlast	1 alive 2 dead
COD	cause of death from death certificate	1 cancer 2 2nd malignancy 3 other - cancer mentioned on DC 4 other=cancer not mentioned on DC 5 Indeterminate (2 primaries present)

STAGE	Clinical stage/substage	1 stage I 2 Ia 3 Ib 4 Ic 5 stage II 6 IIa 7 IIb 8 IIc 9 stage III 10 IIIa 11 IIIb 12 IIIc 13 Stage IV 0 or 99 NK
ADEQ	Adequacy of staging procedures	1 adequate 2 inadequate 0 or 9 NK
HISTO	Histology	1 Serous 2 Mucinous 3 Endometroid 4 Clear cell 5 Germ cell 6 Granulosa 7 Theca 8 Adenocarcinoma 9 Bordeline malignancy 10 Mixed mullerian 11 Brenner 12 Mixed mesodermal 13 Sarcoma 14 Mesonephroid 19 Borderline (serous) 29 Borderline (mucinous) 69 Borderline (granulosa) 0 or 99 NK
GRADE	Tumour grade	1 Well differentiated 2 Moderately differentiated 3 Poorly differentiated 0 or 9 NK
HADSURG	Did patient have surgery	1 No 2 Yes 3 Laparotomy only 0 or 9 NK

TAH OOPH SUBTAH GIT BSO LAVAGE OMENT BIOPSY NODES	Various surgical procedures	Y or blank
SURGEON	Type of surgeon	1 Gynaecological oncologist, (3 surgeons, 1 doing 70 2 Gynaecologist 3 General surgeon 4 Other surgeon or clinician 5 Surgeon outside region or NK 0 or 9 NK
RESDIS	Residual disease	1 None 2 Seedlings 3 < 2cm 4 > 2 cm 5 Bulky 0 or 9 NK
PREVHYST	Previous hysterectomy	1 No 2 Yes 0 or 9 NK
IDS	Intervention debulking surgery	0 no (i.e. second operation after CT) 1 yes
OPTYPE	Extent of surgery	1 Biopsy only 1=dx only, no attempt at surgery 2 GIT surgery +/- palliative surgery 2-5=palliative surgery 3 Oophorectomy (includes prev hysterectomy) 4 Oophorectomy + palliative surgery 5 BSO alone 6 attempted (failed) radical 7-12=radical surgery 7 TAH (+/- ooph, oment) 8 TAH + BSO 9 TAH,BSO, Oment 10 TAH,BSO,Oment, nodes 11 TAH,BSO,Oment, nodes, lavage 12 TAH + GIT 52 BSO + GIT 53 BSO+palliative (group with 5) 54 BSO +palliative++ (group with 5) 0 or 99 No surgery or NK

HADCT	Type of chemotherapy	1 single agent (incomplete data - many blanks) 2 combination chemo blank not known
PM	Diagnosed at post mortem	0 No 1 Yes
OTMALIG	Other malignancy at dx	0 No 1 Concurrent tumour 2 Previous tumour
TYPE	type of tumour	1 endometrial cancer 2 breast cancer 3 colon/rectum 4 cervical cancer (excluding CIN III) 5 Skin cancer 9 miscellaneous
INTERVAL	interval between tumours	0-25+ time in years

Index

Bibliography

[1] C. M. Bishop. Mixture density networks. Technical Report NCRG/94/004, 1994.

[2] C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[3] M. E. Boyd. Ovarian cancer. *The Canadian Journal of Surgery*, 28:No 2:114–118, March 1985.

[4] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley & Sons, 1991.

[5] D. R. Cox. *Renewal Theory.* Methuen, 1962.

[6] M. Lewicki and B. Olshausen. A probabilistic framework for the adaptation and comparison of image codes, 1999.

[7] D. Lowe and A. R. Webb. Exploiting prior knowledge in network optimization: An illustration from medical prognosis. *Network*, 1:299–323, 1990.

[8] D. J. C. MacKay. Bayesian methods for backpropagation networks. In E. Domany, J. L. van Hemmen, and K. Schulten, editors, *Models of Neural Networks III.* Springer-Verlag, New York, 1994.

[9] R. M. Ripley. *Neural Network Models for Breast Cancer Prognosis.* PhD thesis, University of Oxford, 1998.

[10] B. Vincent. Survival prognosis in ovarian cancer, 1999.

[11] A. Webb. *Statistical Pattern Recognition.* Arnold, 1999.