NONLINEAR MODELS FOR NEURAL NETWORKS

Susan Brittain
BSc. (Honours)

Submitted in partial fulfillment of the academic requirements for the degree of

MASTER OF SCIENCE
in
Statistics

In the
School of Mathematics, Statistics and Information Technology
University of Natal
Pietermaritzburg

2000

DECLARATION

The research work described in this dissertation was carried out in the School of Mathematics, Statistics and Information Technology, University of Natal, Pietermaritzburg, under the supervision of Professor L. M. Haines.

I declare that this study represents original work by the author and has not otherwise been submitted in any form for any degree or diploma to any University. Where use has been made of the work of others it is duly acknowledged in the text.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor Professor Linda Haines whose knowledge, patience and support knows no bounds.

I would also like to take this opportunity to thank Barry, Stacy and Megan for all their support and encouragement.

ABSTRACT

The most commonly used applications of hidden-layer feed forward neural networks are to fit curves to regression data or to provide a surface from which a classification rule can be found. From a statistical viewpoint, the principle underpinning these networks is that of nonparametric regression with sigmoidal curves being located and scaled so that their sum approximates the data well, and the underlying mechanism is that of nonlinear regression, with the weights of the network corresponding to parameters in the regression model, and the objective function implemented in the training of the network defining the error structure. The aim of the present study is to use these statistical insights to critically appraise the reliability and the precision of the predicted outputs from a trained hidden-layer feed forward neural network.

Contents

Chapter 1

Introduction

Neural networks were developed primarily by engineers to model data which are intrinsically nonlinear and of high dimension. The main interest was in predictions with very little attention given to inference. Statisticians started to take note of neural networks when it became evident that these networks were carrying out functions that are essentially statistical. For example, hidden-layer feed forward networks are widely used in classification and regression problems. Excellent reviews of neural networks from a statistical perspective can be found in Ripley (1993), Ripley (1996), Cheng and Titterington (1994) and Bishop (1995).

The problem addressed in this thesis is that of setting confidence limits to the predicted responses of a hidden-layer feed forward neural network. If the neural network is regarded as a nonlinear regression model, the environment becomes statistical and

the methodologies from the statistical context can then be adapted and used. Methods for fitting confidence intervals to predicted responses of a nonlinear regression model are however not well developed and are therefore investigated in some depth within the study.

The more specific aim of the thesis is to compare and contrast methods of obtaining confidence intervals to predicted responses for a single hidden-layer feed forward neural network using the linearisation (or Wald), profile likelihood and bootstrap methods. These methods are applied to two examples. The first example is data on bean root cells taken from Ratkowsky (1983), which is known to be close-to-linear in behaviour, and this is used as a benchmark for the second example which consists of artificial data for a single hidden-layer feed forward neural network.

The thesis is divided into five chapters with Chapters 2, 3 and 4 containing the main body of the study. Chapter 2 sets the scene for the thesis and contains an introduction and overview of neural networks, their evolution, the different types of neural networks developed and the application of these networks. Some statistical insights into neural networks are also presented there and problems experienced in the fitting of neural network models described.

In Chapter 3 the nonlinear regression model is introduced and discussed. The problem of fitting confidence intervals to the predicted response is addressed and the three methods of applying confidence intervals; the linearisation (or Wald), profile

likelihood and bootstrap methods, are examined. Each of these methods is developed and discussed in turn, specifically with a view to applying the technique to a single hidden-layer feed forward neural network. In Chapter 4 two examples are introduced, the methodology of Chapter 3 is applied and the results reviewed. The conclusion to this study is presented in Chapter 5. In particular a summary of the three methods of applying confidence intervals to the predicted responses is given and further topics for investigation are briefly discussed.

Chapter 2

Neural Networks

2.1 Introduction

The human brain consists of some 10^{11} to 10^{12} nerve cells known as neurons which are interconnected by nerve fibres to form an intricate network. Neurons are the basic building blocks of the brain and are able to receive, process and transmit impulses or signals over this network resulting in an overall response. Figure 2.1 is a schematic drawing of a typical neuron showing its four main components, the nucleus or cell body, the axon, the dendrites and a synapse. A neuron is very rarely activated by just one other neuron, but rather acts as a "summing amplifier" for the various input signals from a number of other neurons. Nerve impulses are conveyed along the dendrites into the neuron where they are processed by the nucleus. If the sum of the effects of the

impulses causes the electrical potential of the nucleus to reach a certain threshold a pulse is conveyed along the axon and the neuron is said to have "fired", i.e. a nerve impulse is sent along the axon to be transmitted to other connecting neurons. The axon is a single fibre extending from the nucleus which then divides into smaller branches at the end of which are the synapses. A synapse is a gap, measuring approximately 1 millionth of an inch, with a small branch of the axon on the transmitting side of the gap and the dendrite or nucleus of the receiving neuron on the receiving side. It has been established that nerve impulses cross this gap by means of chemical carriers (Wooldridge,1963, pp.5-10; Nathan, 1982, Chapter 9, p.64, Chapter 10, p.72).

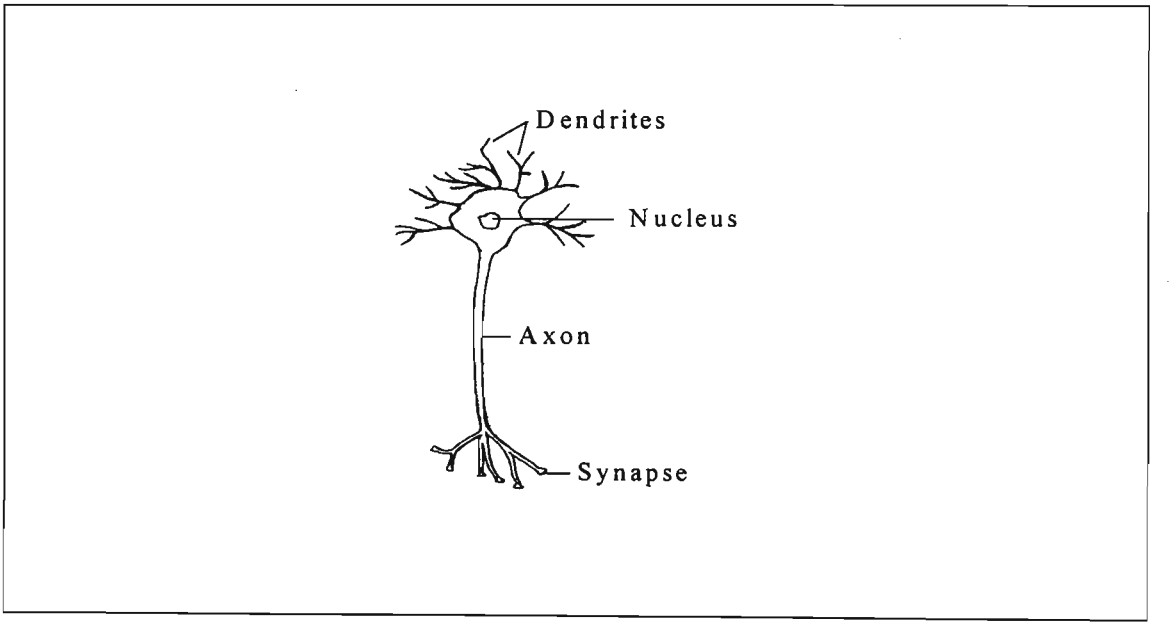

Figure 2.1: A Biological Neuron

CHAPTER 2

Although certain basic structures of the brain are now understood, many features such as the ability of the brain to handle cognitive tasks, including pattern recognition, the understanding of language and the solution of problems by drawing on previous experiences, remain largely unexplained. This is particularly remarkable in the sense that basic operations that take the brain milliseconds to compute require mere nanoseconds by a modern computer. It is the performance of cognitive tasks by the brain that has stimulated the development of artificial neural networks (ANNs). As the biological neuron is the basic building block of the brain, so an artificial neuron is the building block of an ANN. A typical artificial neuron or processing unit is illustrated in Figure 2.2.

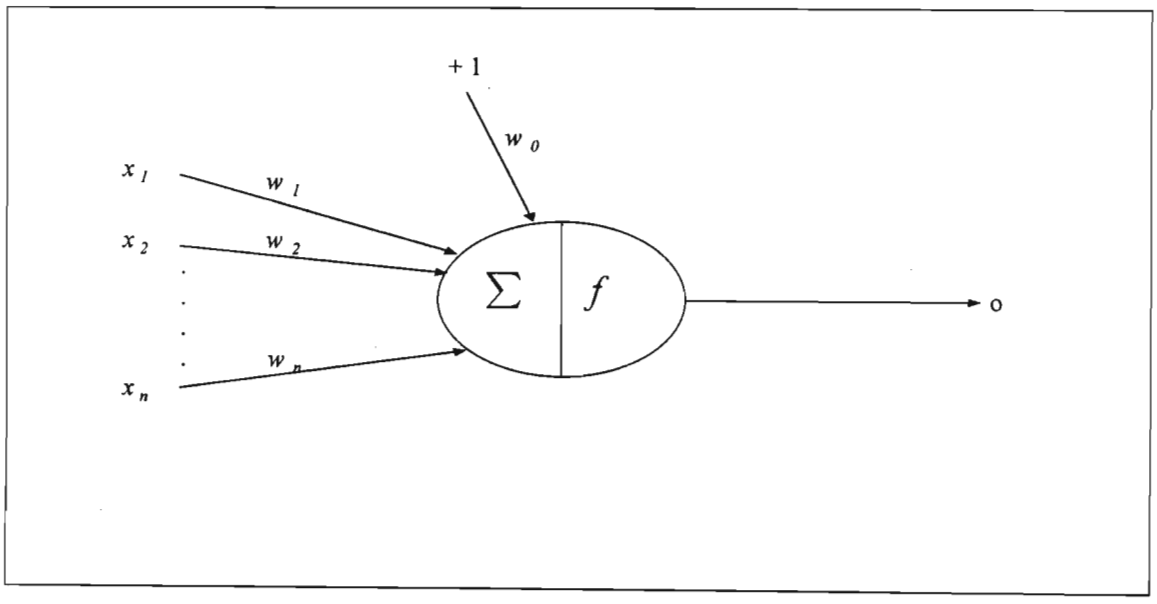

Figure 2.2: An Artificial Neuron

The inputs $x_1, x_2, \ldots, x_n$ are received from an external source or from a set of other artificial neurons and are attenuated by corresponding connecting weights $w_1, w_2, \ldots, w_n$ respectively. The output o is obtained by summing the weighted inputs together with a bias or constant term, w_0, and by applying a transfer or activation function to the resultant sum. This process can therefore be summarised mathematically as:

$$o = f(h) = f(\mathbf{x}^T\mathbf{w} + w_0)$$

where $\mathbf{w} = (w_1, w_2, \ldots, w_n)^T$ and $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ and f is termed the activation or transfer function. Commonly used transfer functions include

- the sign function, $f(h) = \begin{cases} 1 \text{ if } h > 0 \\ -1 \text{ if } h \leq 0 \end{cases}$,
- the logistic function, $f(h) = (1 + e^{-h})^{-1}$, which produces continuous output between 0 and 1,
- the linear function, $f(h) = h$, and
- the Gaussian or radial basis function, $f(h) = e^{-h^2/2}$, with a continuous output between 0 and 1.

The logistic function can in fact be reformulated as the tanh function through the relationship

$$\tanh(\tilde{h}) = \frac{e^{\tilde{h}} - e^{-\tilde{h}}}{e^{\tilde{h}} + e^{-\tilde{h}}} = \frac{2}{(1 + e^{-2\tilde{h}})} - 1 = 2f(2\tilde{h}) - 1$$

where $\tilde{h} = h/2$ and $f(h)$ is the logistic function as described above, with continuous output between -1 and 1 (Bishop, 1995, p.127). These transfer functions are illustrated in Figure 2.3. It should be noted that the functions mentioned above are just a few of the many transfer functions used in the field of ANNs.

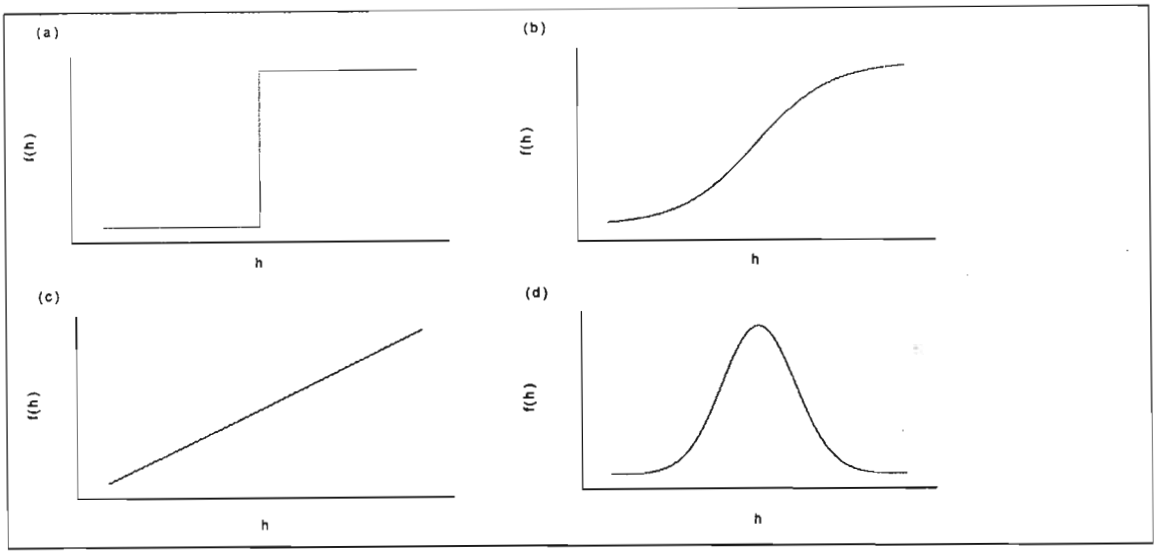

Figure 2.3: (a) the sign transfer function; (b) the logistic transfer function; (c) the linear transfer function; (d) the Gaussian transfer function.

In an attempt to emulate the massive networking capabilities of the brain, artificial neurons are linked together to form a network. There are a number of commonly used network structures which can be divided into three broad classes; hidden-layer feed forward networks, Hopfield networks and self-organising networks and these are considered briefly below.

CHAPTER 2

In 1962 Rosenblatt introduced the simple perceptron, comprising a set of inputs linked to a single layer of artificial neurons with threshold activation functions which produced binary outputs, and demonstrated that this perceptron could be trained to solve a range of input-output problems, specifically classification problems. At the same time Rosenblatt proved the perceptron convergence theorem (Cheng and Titterington, 1994; Bishop, 1995 p.100; Fine, 1999, p.31) which states that the perceptron can be trained to solve linearly separable problems in a finite number of steps. This proof was seen as a major breakthrough in the field of ANNs but in 1969 Minsky and Papert demonstrated that the perceptron could not handle problems such as the "exclusive or" (XOR) problem which are not linearly separable and this caused the research of ANNs to stall for close on twenty years. To obviate this difficulty, perceptrons consisting of more than one layer of neurons with activation functions other than the threshold function, termed hidden-layer feed-forward neural networks or multilayer perceptrons (MLPs), were constructed. The architecture of MLPs was appealing but it was not until 1986, when Rumelhart, Hinton and Williams introduced the backpropagation algorithm, that a method for adjusting the weights of the network so that the network output was in some sense close to a target output was produced. This allowed a greater flexibility in the modelling of data and indeed MLPs have proved particularly useful in classification and regression. In particular, given a set of x-data which are provided as inputs to the MLP, outputs are produced which provide a surface that approximates

a regression function or provides the basis for a cut-off rule for classification. As an example consider the situation depicted in Figure 2.4. A set of input data which belong to one of two classes A or B, represented by $x = 0$ and $x = 1$ respectively, has been fed into an MLP with logistic activation functions. If the output from the network is less than or equal to the cutoff point of 0.5 then the input is classified as A, i.e. close to $x = 0$, and if the output is greater than 0.5 then the input is classified as B, i.e. close to $x = 1$. MLPs have been used in a wide variety of applications including the NETtalk speech generator, zip-code recognition and recognition of sonar targets and continue to be used extensively today.

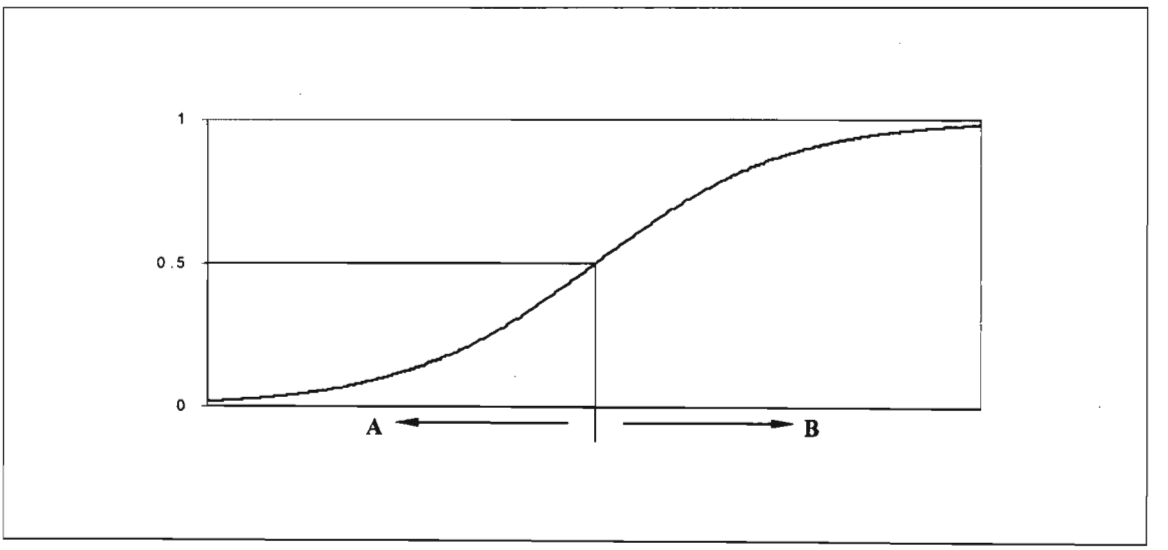

Figure 2.4: An input is classified as type A if the network output is less than 0.5 and as type B if the network output is greater than 0.5

CHAPTER 2

Situations in which the brain uses its perceptive and associative skills are poorly represented by MLPs. In 1982 Hopfield, an American physicist, developed an associative memory ANN termed the Hopfield Network which is widely used today. The success of this network led to a revival of interest in the field of neural networks in the 1980's. The basic Hopfield network comprises interconnected neurons with threshold activation functions. The network is "trained" by feeding into the network a set of p correct patterns, $\mathbf{S}^{(\mu)}$, $\mu = 1, ..., p$, known as exemplars, each comprising n elements, $\mathbf{S}^{(\mu)} = \{S_1^{(\mu)}, S_2^{(\mu)}, \ldots, S_n^{(\mu)}\}$, where $S_i^{(\mu)} = \pm 1$, $i = 1, \ldots, n$. These vectors are used to calculate the weights according to the formula

$$w_{ij} = \frac{1}{n} \sum_{\mu=1}^{p} S_i^{(\mu)} S_j^{(\mu)}, \quad w_{ii} = 0, \quad i, j = 1, \ldots, n. \tag{2.1}$$

The scheme (2.1) is often referred to as the "Hebb rule". To achieve the association of an unknown pattern $\mathbf{x}$ with a particular exemplar, the vector $\mathbf{x}$ is input into the network and the neurons of the network are updated asynchronously, i.e. one at a time, in a deterministic or a random manner according to the scheme

$$S_i = sign(\sum_{j=1}^{n} w_{ij} S_j) \quad i = 1, \ldots, n.$$

This process continues until no further updating can take place and the output of the network, specifying an exemplar, is taken as the pattern which is, in some sense, most closely associated with the input $\mathbf{x}$. The underlying workings of the Hopfield network

can be explained by invoking an energy function given by

$$H = -\frac{1}{2}\sum_{\mu=1}^{p}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}S_i^{(\mu)}S_j^{(\mu)}.$$

It can be shown that the exemplars correspond to local minima of H, often referred to as "basins of attraction", and that the updating process results in a decrease in H so that at termination a local minimum is reached (Hertz, Krogh and Palmer, 1991, pp.21-23; Cheng and Titterington,1994). A problem with Hopfield Networks is the existence of spurious states corresponding to local minima of the energy function H which do not coincide with the given exemplars but various procedures for remedying this situation have been developed. The Hopfield network was seen as an important advance in neural network research and has led to many further developments as for example Boltzmann machines (Hertz, Krogh and Palmer, 1991, Section 7.1, p.163; Ripley, 1993; Ripley, 1996, pp. 279-283).

Multilayer perceptrons and Hopfield networks undergo "supervised" learning in the sense that the network is trained using data sets comprising inputs and target outputs. Self-organising networks are not given target outputs but rather detect features or patterns inherent in the input data, thus displaying a degree of self-organisation. This type of learning is termed "unsupervised" learning. There are two main types of self-organising networks, those that use the Hebbian learning rule and those that use the competitive "winner takes all" rule. Consider a network with n input and m output nodes. In the case of Hebbian learning input vectors $\mathbf{x}^{(\mu)}, \mu = 1, \ldots, p$, are fed into the

network one at a time and the weights, $w_{ij}, i = 1, \ldots n$, $j = 1, \ldots, m$, are adjusted in such a way that the vector of weights, $\mathbf{w}_j$, is just the jth eigenvector corresponding to the jth eigenvalue of the matrix C where $C = \sum_{\mu=1}^{p} \mathbf{x}^{(\mu)}\mathbf{x}^{(\mu)T}$. The output $\mathbf{o}$ therefore represents the first m principal components of the input data $\mathbf{x}$ and the networks are in effect performing principal component analysis (Hertz, Krogh and Palmer, 1991, Chapter 8, p.197). Cluster analysis and Kohonen feature mapping are just two of the applications of competitive or "winner takes all" learning. The simplest competitive learning network comprises a set of binary-valued outputs, $\mathbf{o} = (o_1, \ldots, o_m)$, where o_j represents the jth category, $j = 1, \ldots, m$, each fully connected through the network to the input vector $\mathbf{x}$. Only one output node, the "winner", can be on at a time, and is determined as the node with the largest net output h_j, where $h_j = f(\sum_{i=1}^{n} w_{ij}x_i)$, $j = 1, \ldots, m$, and w_{ij} represents the weights of the connections between the ith input and jth output units. The winning output unit has its output set to 1, and is said to have "fired", while the other output units are set to 0. The input $\mathbf{x}$ is then deemed to be classified as belonging to the jth category. Similar inputs should be classified in the same category and hence should "fire" the same output unit (Hertz, Krogh and Palmer, 1991, Chapter 9, p.217; Ripley, 1993).

The networks mentioned above are just some of the many types of networks that have been developed in the rapidly advancing field of ANNs. The present study will focus on the hidden-layer feed forward neural network or multilayer perceptron.

2.2 Multilayer Perceptrons

2.2.1 Definition

An MLP consists of layers of neurons which are connected by forward links, i.e. there are no feedback loops. Typically such networks consist of an input layer of processing units which accept the individual input values, a number of hidden layers comprising units with the number and the activation functions defined by the user, and an output layer of units corresponding to the required responses. Each link between the neurons has an associated weight. When specifying the number of layers present in the network the input layer is not counted. A typical example of a single hidden-layer MLP is displayed in Figure 2.5 and the output of this network can be developed explicitly as follows. Suppose that there are n input units, h hidden units and m responses. Let $\alpha_{ij}, i = 1, \ldots, n$, $j = 1, \ldots, h$, represent the associated weight between the ith unit in the input layer and the jth unit in the hidden layer. Similarly let β_{jk}, $j = 1, \ldots, h$, $k = 1, \ldots, m$, represent the associated connection between hidden layer unit j and output layer unit k. The output of the jth hidden layer unit is obtained by first forming a weighted linear combination of the n input values and a bias term, denoted α_{0j}, as

$$a_j = \alpha_{0j} + \sum_{i=1}^{n} \alpha_{ij} x_i = \sum_{i=0}^{n} \alpha_{ij} x_i \text{ with } x_0 = 1, \quad j = 1, \ldots, h.$$

CHAPTER 2

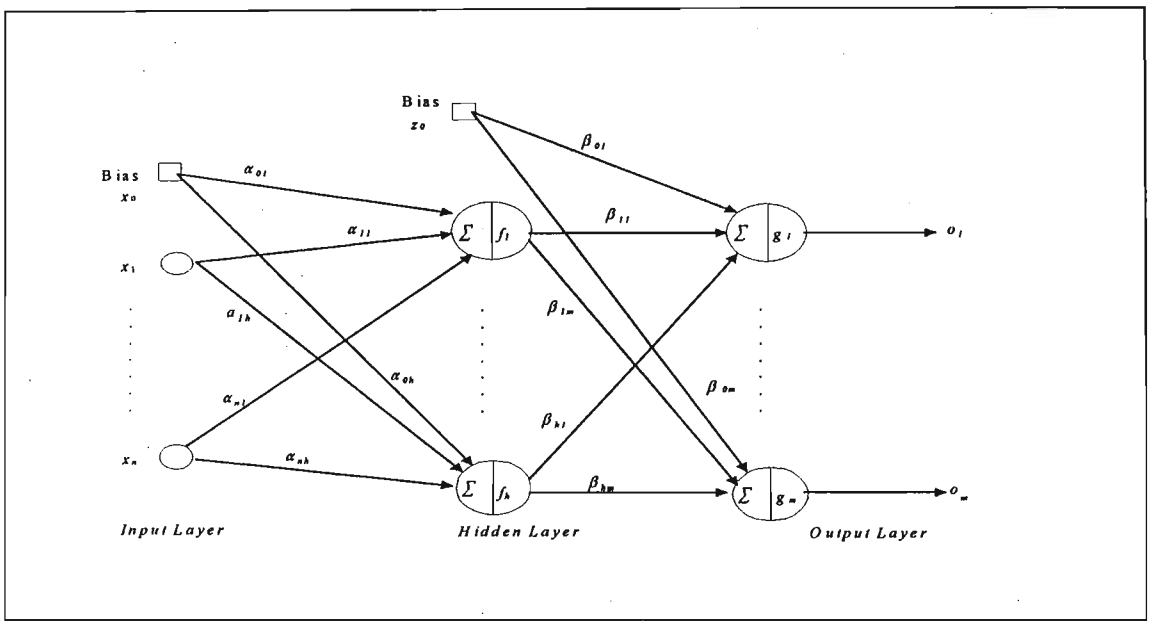

Figure 2.5: A single hidden-layer Multilayer Perceptron (MLP)

The activation or transfer function of the jth hidden unit is applied to this sum to give the output

$$z_j = f_j(a_j), \quad j = 1, \ldots, h.$$

The outputs of the network are produced in a similar manner. Specifically the kth output unit has output

$$o_k = g_k(b_k), \quad k = 1, \ldots, m,$$

where b_k is the weighted linear combination of the outputs of the hidden units and a bias term β_{0k}, and is given by

$$b_k = \beta_{0k} + \sum_{j=1}^{h} \beta_{jk} z_j = \sum_{j=0}^{h} \beta_{jk} z_j \text{ with } z_0 = 1, \quad k = 1, \ldots, m.$$

Overall the output can be summarised as

$$o_k = g_k(\sum_{j=0}^{h} \beta_{jk} f_j(\sum_{i=0}^{n} \alpha_{ij} x_i)), \quad k = 1, \ldots, m.$$

(Bishop, 1995, pp.118-119; Neal, 1996, p. 10; Ripley,1996, pp.143-144; Tibshirani, 1996).

The training data consists of p data sets of the form $\{\mathbf{x}^{(\mu)}, \mathbf{y}^{(\mu)}\}, \mu = 1, \ldots, p$ where the $\{\mathbf{x}^{(\mu)}\}$ terms are the input values and the $\{\mathbf{y}^{(\mu)}\}$ terms are the associated responses or target outputs. The network is presented with one such set of data at a time and the weights of the network are updated such that an objective or error function is minimised. It is this updating of the weights that is interpreted as "training" or "learning". The most commonly used error functions are the sum of squares error function given by

$$E = \sum_{\mu=1}^{p} \sum_{k=1}^{m} (o_k^{(\mu)} - y_k^{(\mu)})^2 \tag{2.2}$$

(Hertz, Krogh and Palmer, 1991, p.117; Cheng and Titterington, 1994; Bishop, 1995, Section 6.1, p.195; Ripley, 1996, p.148) and the cross-entropy error function given by

$$E = -\sum_{\mu=1}^{p} \sum_{k=1}^{m} y_k^{(\mu)} \ln \frac{o_k^{(\mu)}}{y_k^{(\mu)}} \tag{2.3}$$

(Cheng and Titterington, 1994, p.21; Bishop, 1995, Section 6.9 p.237; Ripley, 1996, p.149). The minimisation of the error function with implied updating of the weights is achieved through the use of one of the many optimisation algorithms that are generally available. The selection of the optimisation method is ultimately dependant on the user and includes steepest descent, conjugate gradients, quasi-Newton methods and simulated annealing. Most of these methods are dependant on evaluating the derivatives of the error function and it is the algorithm for finding these derivatives that Bishop (1995, p.140) defines as back-propagation due to the fact that the errors are propagated backwards through the network in order to evaluate the derivatives which are then used to adjust the weights. Once the optimal weights have been established and the error function minimised the network is considered trained. The final test of the network is generalisation, i.e. given new inputs can the network produce good predictions? Good generalisation ultimately means that the network is able to model the true underlying function describing the input data while simultaneously accommodating the noise inherent in the data set. The answer to this question poses several additional questions. These include how to find the optimal architecture of the neural network such that good generaliation is achieved, how to validate and test the network and how to define the training, validation and testing data sets. The optimal architecture of the neural network is covered briefly in the next section. Discussions regarding the formation of the training, validation and testing data sets can be found in Bishop (1995, p.372) and

Fine (1999, pp. 243-245) and will not be mentioned further in this thesis.

A key feature of MLPs is that of the universality property (Bishop, 1995, p.130; Ripley, 1996, p.174) which states that under mild regularity conditions any continuous mapping can be accurately approximated by a network having two layers and logistic activation functions, provided the number of hidden units is sufficiently large. This lends strong theoretical support to the use of MLPs for modeling regression and classification data.

2.2.2 The Bias-variance Dilemma and Overfitting

The main goal of using a neural network is to learn from a given data set and to use this information to generalise to new inputs. Poor generalisation is a product of an inadequate network and is commonly due to the fact that it is extremely difficult to establish the ideal number of units in each of the hidden layers in an MLP. Rosenblatt's perceptron is an example with no hidden units and this leads to its inability to fit nonlinearly separable functions. At the other extreme is the case of an MLP with a very large number of hidden units which can lead to an almost exact fitting of the training data, known as overfitting, but very poor generalisation properties (Bishop, 1995, p.332). Thus a compromise number of hidden units is sought in order to determine the structure of the MLP that attains the best generalisation possible. The bias-variance dilemma offers an insight into the complexity of this problem.

CHAPTER 2

Bias is a measure of the average "distance" between the true value of a function and its estimate. In the case of neural networks this amounts to a measure of the difference between the true output or target values and that provided by the network. Obviously the closer the network output is to the true target values the smaller the bias will be and conversely the further away they are, the larger the bias will be. Variance, on the other hand, is a measure of the average "distance" between an estimated function and its expected value. In the context of neural networks this is equivalent to the distance between the network output and the expected output of that particular network. Variance is therefore extremely sensitive to the particular data set undergoing training. It is usually very unlikely that a network will exhibit a small bias and a small variance. These two quantities are complementary in the sense that a small bias usually results in a large variance and vice-versa. Thus a compromise is sought so that both the bias and the variance are reasonably small. This is known as the bias-variance dilemma. Geman, Bienenstock and Doursat (1992) quantified this dilemma by showing that the error as in (2.2) or (2.3) can be decomposed into the bias squared plus the variance. In particular, suppose the true value of the underlying function generating the data set is $h(\mathbf{x})$. Suppose further that y is modelled using an MLP as $y = o + \epsilon$ and that the estimated output from this model is $\hat{o}(\mathbf{x})$. Then the mean square error of the estimated responses can be expressed as

$$E_D[(h(\mathbf{x}) - \hat{o}(\mathbf{x}))^2] = E_D[\hat{o}(\mathbf{x}) - (E_D[\hat{o}(\mathbf{x})])^2] + \{E_D[\hat{o}(\mathbf{x})] - h(\mathbf{x})\}^2 \qquad (2.4)$$

where the subscript D refers to the expectation with respect to the MLP model. The first term on the right hand side of (2.4) is the variance of the estimated response and the second term is the bias of the expected value of that response squared (Bishop, 1995, Section 9.1, pp. 333-335). From expression (2.4) it can be seen that decreasing the bias results in an increase in variance and vice-versa. Variance, and hence the expected error, can be reduced by removing some hidden units of the network, but there is the danger that this will increase the bias which in turn will result in the expected error increasing. In order to alleviate this dilemma more data points should be added to the training set but this is not always possible.

Methods for controlling the complexity of the network have been developed in order to control the problem of overfitting. One such technique is termed regularisation and involves adding a penalty to the error function (Ripley, 1993; Cheng and Titterington, 1994; Bishop, 1995, p.338-343; Ripley, 1996, p.157; Fine, 1999, p.220). This penalty function is such that a large number of hidden units will incur a large penalty whereas a small number of hidden units will result in a small penalty. The simplest such regulariser is weight decay and involves minimising the composite function

$$\tilde{E} = E + \frac{\xi}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{h}\alpha_{ij}^2 + \sum_{j=1}^{h}\sum_{k=1}^{m}\beta_{jk}^2\right)$$

where E is as defined in (2.2) or (2.3) and ξ is a controlling parameter balancing the fitting of the MLP and the effect of the penalty function. In statistical terms this approach is equivalent to ridge regression. The technique is also useful in that it

stabilises the solutions of the optimisation algorithm numerically. Another commonly used method for controlling overfitting is that of early stopping where training of the MLP is stopped at the point where the error is a minimum for a separate validation set (Bishop, 1995, p.343). Usually the data set under consideration is split into a training, a test and a validation set which is an inefficient use of data especially when the sample size is small. This also leads to questions regarding how the data should be split. An optimal number of hidden units can also be achieved by either growing or pruning a network (Ripley, 1993; Bishop, 1995, p.353; Ripley, 1996, p.169; Fine, 1999, pp. 232-234). As the names suggest, growing a network is the procedure of starting with a small number of hidden units and then adding units one at a time, while, in contrast, pruning is the process in which a complex network is constructed initially and then connections and units are systematically removed.

A totally different approach is to use Bayesian methods of fitting the underlying model but these techniques will not be considered in this study.

2.2.3 Statistical Insights

The MLP as discussed above is a framework for fitting a weighted sum of activation functions of input data, according to the number of hidden units, to produce an approximate model for the training data. To simplify matters the case of an MLP with a single input and single output unit will be considered in the remainder of this text.

CHAPTER 2

Consider a single hidden-layer feed forward network with two hidden units each with logistic transfer functions used to model regression data of the form $\{x_i, y_i\}, i = 1, \ldots, n$. A single input, x, is fed into the network with a bias term and a single output, o, is produced via a neuron with a linear transfer function. This output can be written explicitly as

$$o = \theta_5 + \frac{\theta_6}{1 + e^{-(\theta_1 + \theta_2 x)}} + \frac{\theta_7}{1 + e^{-(\theta_3 + \theta_4 x)}},$$

where $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_7)^T$ is the vector of unknown connection weights. This network is illustrated in Figure 2.6. If the network is trained by minimising the sum of squares

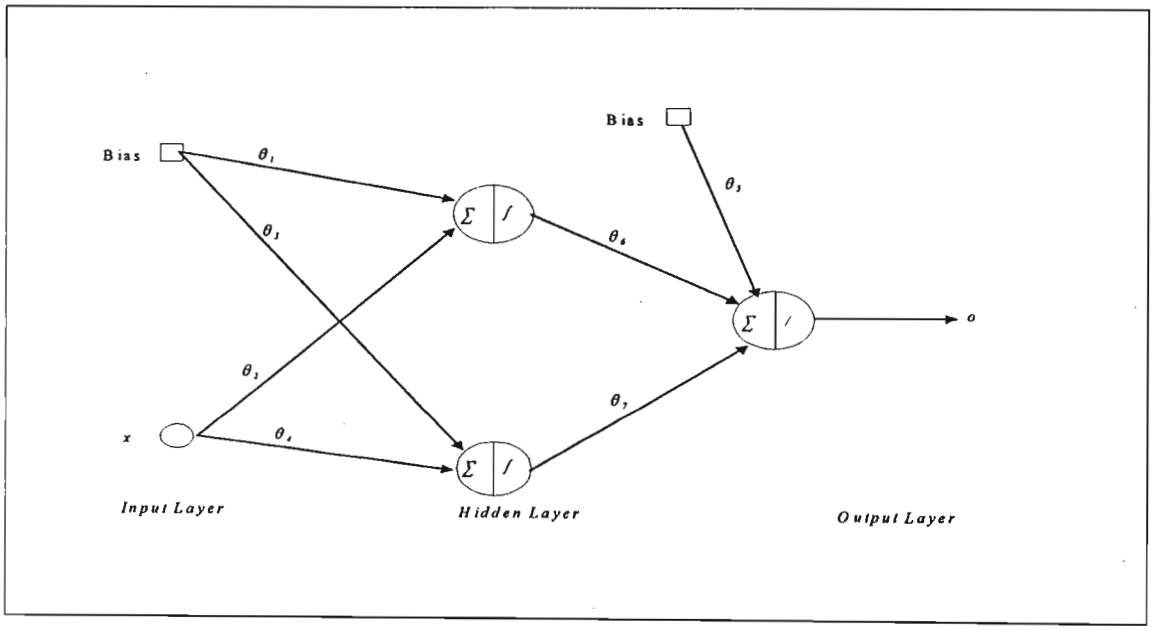

Figure 2.6: A single hidden-layer MLP containing two hidden nodes with logistic activation functions.

error function, $S = \sum_{i=1}^{n}(y_i - o_i)^2$, then the process is, in essence, one of fitting a nonlinear regression model

$$y_i = o_i + \epsilon_i, \quad i = 1, \ldots, n, \tag{2.5}$$

where the error terms, ϵ_i, are independently and identically distributed with mean zero and constant variance, σ^2, to the data. The connection weights are equivalent to the parameters in the regression model, the training of the network is analogous to the iterations in an appropriate optimisation algorithm for minimising the sum of squares error with respect to the parameters and the generalisation of the network corresponds to the prediction of new output values. In the context of neural networks the nonlinear regression function in (2.5) has no real meaning in relation to the data in the sense that it is the function o that is approximating the true output and the weights are just artifacts of this process. Thus the modelling procedure can be viewed as summing scaled and located logistic functions which together with a constant term approximate a smooth curve and this in turn approximates the true output. Figure 2.7 illustrates two logistic curves plus a constant term and the smooth curve that corresponds to their sum. The flexibility apparent in this model indicates that the underlying model of the network is ultimately a nonparametric regression model and in fact misspecifies the true model (Brittain and Haines, 1997).

In general MLP's are widely used to model regression data and in the area of classification (Ripley, 1993; Cheng and Titterington, 1994; Bishop, 1995, p.116). It is

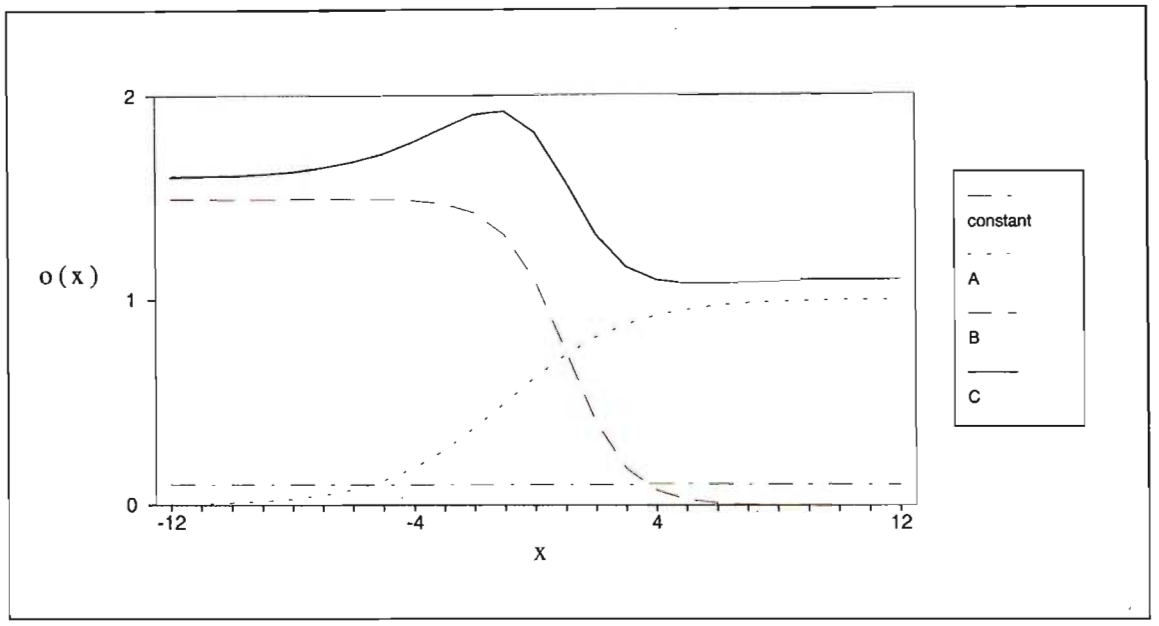

Figure 2.7: A represents a logistic function with a positive slope, B represents a logistic function with a negative slope and C represents the sum of these two functions plus a constant term

also interesting to note that an MLP with linear activation functions corresponds to a multiple regression model.

2.2.4 Problems

ANN's have been developed primarily by engineers who use biological concepts to improve existing and create new models and by neurophysiologists who are investigating the brain and its computing capabilities. Investigations into ANN's by the engineers

have tended to ignore the statistical aspects of these networks and have been restricted to a "black box" scenario. In essence users of ANN's have been concerned with the output of the network and the ability of the network to generalise. As mentioned above, there is concern with regards to the overfitting of the model and to the predictive properties of the network.

Statisticians are interested in inferences and specifically, in the present context, in inferences that can be drawn from fitting regression models. In particular the statistician is concerned with measures of confidence and error for the parameter estimates and for the predicted responses, but usually places most emphasis on the parameter estimation. The aim of this study is therefore to concentrate on the predicted responses of an MLP and, in particular, the errors associated with these responses, since this is an area that has been neglected in the many publications on neural networks. During the course of this study Hwang and Ding (1997) produced a paper which investigates linearised confidence intervals for predicted responses from a neural network as did De Veaux, Schumi, Schweinsberg and Ungar (1998) who also looked at linearised prediction intervals as well as using early stopping and weight decay as alternative methods of constructing prediction intervals. The model describing an MLP is unusual in the statistical context in that it is a sum of scaled logistic functions and as a consequence is highly overparameterised and possibly exhibits multicollinearity. The statistical insights provided in this section, particularly those relating to nonlinear re-

gression, are invoked in order to construct confidence limits for the predicted responses of an MLP and there is a wealth of tools available in the theory of nonlinear regression to tackle such a problem. Inferences for predicted responses in an MLP is the topic of investigation in the remainder of this thesis.

Chapter 3

The Nonlinear Regression Model

3.1 Introduction

Much of the work done in the area of nonlinear regression concentrates on parameter estimation in the nonlinear model, usually due to the fact that these parameters have a specific meaning for the problem to which the model is applied. Matters of concern are the estimation of the parameters and the accuracy of the resultant estimates as measured by their standard errors. A parameter estimate together with its associated standard error can be used to find confidence limits for the corresponding true parameter value. In the case of neural networks the parameters have no meaning and are thus of no particular interest. The main emphasis of the neural network is to produce an output which is analogous to the predicted response from a nonlinear regression model.

CHAPTER 3

Thus a multilayer perceptron (MLP) can be cast as a nonlinear regression model, as discussed in Chapter 2, and the techniques applicable to nonlinear regression can be borrowed and utilised to set confidence intervals to the output. These techniques are the topic of discussion in this chapter.

Consider a nonlinear regression model given by

$$y_i = \eta(x_i, \boldsymbol{\theta}) + \epsilon_i, \ i = 1, \ldots, n, \tag{3.1}$$

where y_i is the observed value at x_i, $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_p)$ is a $p \times 1$ vector of unknown parameters, $\eta(.,.)$ is a nonlinear function and the error terms, ϵ_i, are independently and identically distributed (i.i.d.) with mean zero and constant variance, σ^2. The most common method of obtaining the parameter estimates is the least squares method. Specifically, the least squares estimate of $\boldsymbol{\theta}$, denoted by $\hat{\boldsymbol{\theta}}$, is obtained by minimising the error sum of squares,

$$S(\boldsymbol{\theta}) = \sum_{i=1}^{n} [y_i - \eta(x_i, \boldsymbol{\theta})]^2 \tag{3.2}$$

with respect to $\boldsymbol{\theta}$. Using the least squares estimate, $\hat{\boldsymbol{\theta}}$, in place of $\boldsymbol{\theta}$ in $S(\boldsymbol{\theta})$ above and dividing by the appropriate degrees of freedom provides an estimate of the unknown error variance, σ^2, as

$$s^2 = \frac{S(\hat{\boldsymbol{\theta}})}{n - p}. \tag{3.3}$$

The minimisation of (3.2) can result in a local instead of the global minimum, $\hat{\boldsymbol{\theta}}$, and as a consequence, a large amount of research on parameter estimation has concentrated

on the algorithms used to minimise (3.2), the mostly commonly used being the Gauss-Newton technique and its variants. The parameter estimates can also be found by means of the method of maximum likelihood, resulting in an estimate of the unknown error variance, σ^2, as

$$\hat{\sigma}^2 = \frac{S(\hat{\boldsymbol{\theta}})}{n}$$

where $\hat{\boldsymbol{\theta}}$ is the maximum likelihood estimate of $\boldsymbol{\theta}$. If it is assumed that the error terms, ϵ_i, $i = 1, \ldots, n$, are normally distributed then the maximum likelihood estimator of $\boldsymbol{\theta}$ is equal to the least squares estimate of $\boldsymbol{\theta}$. If interest lay solely in the parameter estimates, then the next step would be to construct confidence intervals for $\boldsymbol{\theta}$ using either linearisation, likelihood or resampling methods. However, interest here focuses on a nonlinear function of $\boldsymbol{\theta}$, the mean predicted response $\eta(x_g, \boldsymbol{\theta})$ for a particular value of x, x_g, and the concepts relating to confidence intervals for $\boldsymbol{\theta}$ are extended to this case. There has been surprisingly little research on the problem of constructing confidence limits for predicted values and indeed only Clarke (1987), Vecchia and Cooley (1987), Seber and Wild (1989, p. 192 and p. 235) and Tibshirani (1996) have addressed this problem. The aim in the present study is to concentrate on three specific methods of constructing confidence intervals for the mean predicted value, namely the linearisation method, the profile likelihood method and the bootstrap method, each of which is described in detail in this chapter. The application of these methods to two specific examples follows in Chapter 4.

3.2 Linearisation Method

The linearisation method, also known as the Wald or delta method, is probably the most widely used method for obtaining confidence intervals for the parameters of a model and for functions of those parameters. A description of this technique can be found in most textbooks on linear or nonlinear modelling and one of the most comprehensive treatments within the nonlinear context is provided by Seber and Wild (1989, p.23 and p.192).

Under certain regularity conditions $\hat{\boldsymbol{\theta}}$ and s^2 are consistent estimators of $\boldsymbol{\theta}$ and σ^2 respectively (Seber and Wild, 1989, p.564) and if further regularity conditions are specified then, for large sample sizes, $\hat{\boldsymbol{\theta}}$ is approximately normally distributed with mean $\boldsymbol{\theta}$ and variance

$$V(\hat{\boldsymbol{\theta}}) \approx \sigma^2[\sum_{i=1}^{n} g(x_i, \boldsymbol{\theta})g(x_i, \boldsymbol{\theta})^T]^{-1}_{\hat{\boldsymbol{\theta}}} \tag{3.4}$$

where

$$g(x_i, \boldsymbol{\theta}) = \partial\eta(x_i, \boldsymbol{\theta})/\partial\boldsymbol{\theta}, i = 1, \ldots, n$$

and the subscript $\hat{\boldsymbol{\theta}}$ denotes evaluation at that point (Donaldson and Schnabel, 1987; Seber and Wild, 1989, p.24 and p.568). This result is obtained by taking the first order Taylor expansion of $\eta(x_i, \boldsymbol{\theta})$ about $\hat{\boldsymbol{\theta}}$. If the maximum likelihood method is used to obtain $\hat{\boldsymbol{\theta}}$ then the same result is obtained by taking the inverse of the expected

information matrix evaluated at $\hat{\boldsymbol{\theta}}$,

$$V(\hat{\boldsymbol{\theta}}) \approx E\left[-\frac{\partial^2 l}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T}\right]^{-1}_{\hat{\boldsymbol{\theta}}} = \sigma^2[\sum_{i=1}^{n} g(x_i,\boldsymbol{\theta})g(x_i,\boldsymbol{\theta})^T]^{-1}_{\hat{\boldsymbol{\theta}}},$$

where l is the log-likelihood function of the relevant normal distribution (Seber and Wild, 1989, pp.32-34). An asymptotic $100(1-\alpha)\%$ confidence region for $\boldsymbol{\theta}$ can be expressed as

$$(\hat{\boldsymbol{\theta}}-\boldsymbol{\theta})^T[\sum_{i=1}^{n} g(x_i,\boldsymbol{\theta})g(x_i,\boldsymbol{\theta})^T]_{\hat{\boldsymbol{\theta}}}(\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}) \leq ps^2F_{p,n-p,\alpha}$$

where $F_{p,n-p,\alpha}$ is the appropriate critical F value with p and $n-p$ degrees of freedom and s is as defined in (3.3). In addition a $100(1-\alpha)\%$ confidence interval for a particular parameter, θ_r, is given by

$$\hat{\theta}_r \pm t_{n-p,\frac{\alpha}{2}}\, s\{[\sum_{i=1}^{n} g(x_i,\boldsymbol{\theta})g(x_i,\boldsymbol{\theta})^T]^{-1/2}_{\hat{\boldsymbol{\theta}}}\}^{rr}$$

where $t_{n-p,\frac{\alpha}{2}}$ is the requisite critical t value with $n-p$ degrees of freedom, and the superscript rr refers to the rth diagonal element of the matrix

$$[\sum_{i=1}^{n} g(x_i,\boldsymbol{\theta})g(x_i,\boldsymbol{\theta})^T]^{-1/2}_{\hat{\boldsymbol{\theta}}}.$$

The construction of a $100(1-\alpha)\%$ confidence interval for $\eta(x_g,\boldsymbol{\theta})$ is based on a first order Taylor expansion of the nonlinear function $\eta(.,.)$ and the approximation of the variance of the parameter estimates $\hat{\boldsymbol{\theta}}$ given in (3.4). Specifically for the nonlinear function $\eta(x_g,\hat{\boldsymbol{\theta}})$ the first order Taylor expansion of $\eta(x_g,\hat{\boldsymbol{\theta}})$ about $\boldsymbol{\theta}$ is

$$\eta(x_g,\hat{\boldsymbol{\theta}}) \approx \eta(x_g,\boldsymbol{\theta}) + g(x_g,\boldsymbol{\theta})]_{\boldsymbol{\theta}}\,(\hat{\boldsymbol{\theta}}-\boldsymbol{\theta})$$

which implies that

$$V\left[\eta(x_g,\hat{\theta})\right] \approx V\{g(x_g,\theta)]_\theta\,(\hat{\theta}-\theta)\}$$

$$= g(x_g,\theta)]_\theta^T\, V(\hat{\theta})\; g(x_g,\theta)]_\theta$$

where $V(\hat{\theta})$ is as specified in (3.4). Using the fact that $\hat{\theta}$ is approximately normal with mean θ and variance $V(\hat{\theta})$ it then follows that $\eta(x_g,\hat{\theta})$ is approximately normal with mean $\eta(x_g,\theta)$ and variance

$$g(x_g,\theta)]_\theta^T\, V(\hat{\theta})\; g(x_g,\theta)]_\theta$$

and hence that an approximate $100(1-\alpha)\%$ confidence interval for $\eta(x_g,\theta)$ can be constructed as

$$\eta(x_g,\hat{\theta})\pm t_{n-p,\frac{\alpha}{2}}\, s\,\sqrt{g(x_g,\theta)]_{\hat{\theta}}^T\,[\sum_{i=1}^{n} g(x_i,\theta)g(x_i,\theta)^T]_{\hat{\theta}}^{-1}\; g(x_g,\theta)]_{\hat{\theta}}} \qquad (3.5)$$

(Ratkowsky,1986, p.186; Bates and Watts, 1988, pp. 58-59; Seber and Wild, 1989, pp. 192-193).

The main advantage to using the linearisation method is that it is a quick and easy means of obtaining confidence intervals and that these intervals are readily understood. For this reason the method is the preferred method in many statistical packages. The main disadvantage is that the linearised confidence limits can be entirely meaningless if the normal approximation is poor. In the case where the normal approximation is good the distribution of the parameter estimate under consideration will be

close to symmetric and this will be reflected in the linearised confidence limits which are themselves symmetric. However, in the case where the normal approximation is unsatisfactory, the distribution of the parameter estimate may well be asymmetric and the linearised confidence limits, being necessarily symmetric, will not reflect this asymmetry. Donaldson and Schnabel (1987) compared confidence intervals constructed by means of the lack-of-fit (exact), linearisation and likelihood methods empirically using coverages. Their conclusion was that the intervals calculated using the linearisation method can perform extremely badly compared with the other two methods. They cite a particular case where an observed coverage of 75.0% was obtained for a nominal 95% coverage, although they do acknowledge that the linearisation method was by far the simplest technique to implement. Donaldson and Schnabel (1987) also investigated how three variants of $V(\hat{\boldsymbol{\theta}})$ influenced the observed coverages of the confidence intervals determined by the linearisation method. The three variants investigated were

$$\hat{V}_a = s^2[\sum_{i=1}^{n} g(x_i, \boldsymbol{\theta}) g(x_i, \boldsymbol{\theta})^T]^{-1}$$

as in (3.4),

$$\hat{V}_b = s^2 H(\hat{\boldsymbol{\theta}})^{-1}$$

where $H(\hat{\boldsymbol{\theta}})$ is the Hessian matrix of $S(\boldsymbol{\theta})$ at $\hat{\boldsymbol{\theta}}$, and

$$\hat{V}_c = s^2 H(\hat{\boldsymbol{\theta}})^{-1} (\sum_{i=1}^{n} g(x_i, \boldsymbol{\theta}) g(x_i, \boldsymbol{\theta})^T) H(\hat{\boldsymbol{\theta}})^{-1}.$$

Their conclusion was that there appeared to be no major difference between the three

variants and hence that (3.4) is a satisfactory estimate of the variance of $\hat{\boldsymbol{\theta}}$ to use when constructing confidence regions and confidence intervals as it is "simpler, less expensive, and more numerically stable to compute" (Donaldson and Schnabel, 1987).

Donaldson and Schnabel (1987) also showed that the measures of curvature developed by Bates and Watts (1988) are useful for determining when a linearisation confidence interval will be poor. The solution locus, or expectation surface, for a non-linear model is defined to be the surface generated by the expected responses $\eta(x_i, \boldsymbol{\theta})$, $i = 1, \ldots, n$, in $n-$dimensional space for all possible values of $\boldsymbol{\theta}$. Bates and Watts (1988) defined the *intrinsic curvature* (IN) as a measure of the degree to which the expectation surface deviates from planarity as $\boldsymbol{\theta}$ changes near $\hat{\boldsymbol{\theta}}$, and is inherent in the structure of the data together with the model under consideration. A coordinate grid of $\boldsymbol{\theta}$ values projected onto the expectation surface can be constructed and the *parameter effects curvature* (PE) measures the extent to which this grid is non uniform and curved. In the case of a linear model both the IN and the PE curvature measures are zero, while these measures are nonzero in the case of nonlinear functions. The PE curvature measure is dependent on the parameterisation of the model and can therefore be reduced through reparameterisation of the model. According to Bates and Watts (1988) when the PE measure is small compared to the critical value $1/\sqrt{F_{p,n-p,\alpha}}$ then the assumption that the coordinate grid is approximately linear is valid. Similarly if IN is small compared to $1/\sqrt{F_{p,n-p,\alpha}}$ then it can be assumed that the solution locus is close

to planar. Donaldson and Schnabel (1987) showed that small values of PE indicate a good approximation by the linearisation method while large values suggest a poor approximation. Hence the curvature associated with a nonlinear model and in particular the PE curvature measure should be investigated fully before the linearisation method is used.

3.3 Profile Likelihood Method

Confidence regions obtained by the linearisation method are not always reliable (Donaldson and Schnabel, 1987) and for this reason likelihood-based confidence regions and intervals have been extensively investigated. A $100(1-\alpha)\%$ likelihood-based confidence region for $\boldsymbol{\theta}$ is defined as all values of $\boldsymbol{\theta}$ such that

$$S(\boldsymbol{\theta}) - \mathbf{S}(\hat{\boldsymbol{\theta}}) \leq s^2 p F_{p,n-p,\alpha} \tag{3.6}$$

where $F_{p,n-p,\alpha}$ is the appropriate critical F value with p and $n-p$ degrees of freedom (Donaldson and Schnabel, 1987; Bates and Watts, 1988, p.201; Seber and Wild, 1989, p.98). This confidence region is in fact defined by contours of equal likelihood which are often distorted and ill-defined and cannot, in any case, be visualised when the number of parameters, p, is greater than 2 (Bates and Watts, 1988, p.204). A set of $100(1-\alpha)\%$ simultaneous confidence intervals, one for each parameter θ_r, $r = 1, \ldots, p$,

can be obtained by the Bonferroni method as

$$\hat{\theta}_r \pm t_{n-p,\frac{\alpha}{2p}}\, s\, \{[\sum_{i=1}^{n} g(x_i,\boldsymbol{\theta})g(x_i,\boldsymbol{\theta})^T]^{-1/2}_{\hat{\boldsymbol{\theta}}}\}^{rr},$$

where $t_{n-p,\frac{\alpha}{2p}}$ is the appropriate critical t value with $n-p$ degrees of freedom (Seber and Wild, 1989, p.192), or more conservatively by Scheffe's method as

$$\hat{\theta}_j \pm [pF_{p,n-p,\alpha}]^{1/2}\, s\, \{[\sum_{i=1}^{n} g(x_i,\boldsymbol{\theta})g(x_i,\boldsymbol{\theta})^T]^{-1/2}_{\hat{\boldsymbol{\theta}}}\}^{rr}$$

(Seber and Wild, 1989, p. 194). Other likelihood methods developed include the profile likelihood method of determining a confidence interval for a parameter of interest in a specified model and this approach is used in the present study. The profile likelihood has generated a substantial amount of interest in the statistical literature and discussions concerning this method for a single model parameter can be found in Aitken (1982), Cox and Reid (1987), Ritter and Bates (1993) and Ritter, Bisgaard and Bates (1994).

The profile likelihood of an individual parameter, which is an element of the unknown parameter vector $\boldsymbol{\theta}$ describing a distribution, is constructed as follows. Consider the joint probability distribution of n observations, $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, denoted by $f(\mathbf{x};\boldsymbol{\theta})$, as a likelihood function denoted by $L(\boldsymbol{\theta}|\mathbf{x})$, a function of the unknown parameters $\boldsymbol{\theta}$ for fixed $\mathbf{x}$. Let $l(\boldsymbol{\theta}|\mathbf{x}) = ln(L(\boldsymbol{\theta}|\mathbf{x}))$ denote the log likelihood function. Suppose that $\hat{\boldsymbol{\theta}}$ maximises $l = l(\boldsymbol{\theta}|\mathbf{x})$, i.e. $\hat{\boldsymbol{\theta}}$ is the maximum likelihood estimate (m.l.e.) of $\boldsymbol{\theta}$, and write $\hat{l} = l(\hat{\boldsymbol{\theta}}|\mathbf{x})$. A confidence interval for a particular parameter, θ_j say, is obtained by fixing θ_j at a specific value θ_j^c and then maximising $l(\boldsymbol{\theta}_{(-j)}|\theta_j^c, \mathbf{x})$ with re-

spect to $\boldsymbol{\theta}_{(-j)}$, where $\boldsymbol{\theta}_{(-j)}$ represents the vector $\boldsymbol{\theta}$ excluding the fixed parameter θ_j. The estimates of $\boldsymbol{\theta}_{(-j)}$ so obtained are denoted by $\hat{\hat{\boldsymbol{\theta}}}_{(-j)} \mid \theta_j^c$. This process is repeated for values of $\theta_j = \hat{\theta}_j + m\delta$, where $m = \pm 1, \pm 2, \ldots$, and δ is a selected step size resulting in a function of θ_j expressed as $\hat{\hat{l}}(\theta_j) = l(\hat{\hat{\boldsymbol{\theta}}}_{(-j)} \mid \theta_j, \mathbf{x})$ which can then be plotted against the θ_j values to produce a curve depicting the profile log likelihood. A $100(1-\alpha)\%$ confidence interval for θ_j is given by all values of θ_j satisfying

$$\hat{\hat{l}}(\theta_j) - \hat{l} \leq s^2 \chi^2_{p,\alpha} \tag{3.7}$$

where $\chi^2_{p,\alpha}$ is the appropriate critical χ^2 value with p degrees of freedom and the actual confidence limits are determined by the points of intersection between the curve $l(\hat{\hat{\boldsymbol{\theta}}}_{(-j)} \mid \theta_j)$ and the horizontal line $l(\hat{\boldsymbol{\theta}}) + s^2\chi^2_{p,\alpha}$. The number of points plotted is dependent on the step size, δ, and the inequality in (3.7). Cook and Weisberg (1990) suggest using a step size of $\delta = 0.2 \times se(\hat{\theta}_j)$ to start with and then repeatedly halving δ in cases where the parameter estimates fail to converge, but they indicate that a fixed step size is not always ideal. Cook and Weisberg (1990) found that in cases where the profile log likelihood is close to quadratic too many evaluations of $\hat{\hat{l}}(\theta_j) = l(\hat{\hat{\boldsymbol{\theta}}}_{(-j)} \mid \theta_j)$ tend to take place, while in cases where the profile log likelihood is skewed the step sizes may be too large to observe the true nature of the curve. To remedy this, Cook and Weisberg (1990) present a method whereby a dynamic step size based on the curvature of the profile log likelihood function at the current value of θ_j can be determined.

The bisection method can be used to determine the points of intersection

described by the equality in (3.7). An excellent description of the bisection method and a FORTRAN routine for the algorithm can be found in Press, Flannery, Teukolsky and Vetterling (1986).

For the normal distribution the log likelihood function is directly proportional to the error sum of squares and hence, for the nonlinear model (3.1), maximising the likelihood is equivalent to minimising the error sum of squares $S(\boldsymbol{\theta})$, defined in (3.2). Thus the ideas developed above for the profile log likelihood function can immediately be adapted to working with $S(\boldsymbol{\theta})$ and the derivation developed above is equivalent to minimising $S(\boldsymbol{\theta}_{(-j)}|\theta_j^c)$ with respect to $\boldsymbol{\theta}_{(-j)}$, for fixed $\theta_j = \theta_j^c$, resulting in the parameter estimates $\hat{\hat{\boldsymbol{\theta}}}_{(-j)} \mid \theta_j^c$, as before, and the corresponding sum of squares $S(\hat{\hat{\boldsymbol{\theta}}}_{(-j)} \mid \theta_j^c)$. When using the sum of squares approach (3.7) is rewritten as

$$S(\hat{\hat{\boldsymbol{\theta}}}_{(-j)} \mid \theta_j^c) - S(\hat{\boldsymbol{\theta}}) \leq s^2 t^2_{n-p,\frac{\alpha}{2}}. \tag{3.8}$$

It is more common to find the sum of squares approach rather than the full likelihood function being used in the construction of profile likelihoods (Donaldson and Schnabel, 1987; Bates and Watts, 1988, p. 201; Cook and Weisberg, 1990).

For a linear model, i.e. $\eta(x_i, \boldsymbol{\theta})$ is linear in the parameters $\boldsymbol{\theta}$, the sum of squares function $S(\boldsymbol{\theta})$ is a quadratic function in $\boldsymbol{\theta}$ and it follows immediately that the function $S(\hat{\hat{\boldsymbol{\theta}}}_{(-j)} \mid \theta_j)$ will be quadratic in θ_j. Thus the profile log likelihood curve is itself a quadratic. In general the profile log likelihood of a parameter in a nonlinear model such as (3.1) will be expected to deviate from a quadratic curve according to

the degree of nonlinearity in the model, i.e. the larger the nonlinearity, specifically the PE curvature, the more skewed the profile log likelihood will appear (Donaldson and Schnabel, 1987; Bates and Watts, 1988, p. 205; Cook and Weisberg, 1990). There are, however, exceptions to this and some are detailed in Donaldson and Schnabel (1987) and Cook and Weisberg (1990).

A related approach to the profile likelihood method is the profile t plot described in Bates and Watts (1988, Section 6.1.2, pp. 205-206). A profile t plot of the parameter of interest, θ_j, comprises a plot of $\tau(\theta_j) = sign(\theta_j - \hat{\theta}_j)\sqrt{S(\hat{\hat{\boldsymbol{\theta}}}_{(-j)} \mid \theta_j) - S(\hat{\boldsymbol{\theta}})}/s$ on the y-axis and the studentised parameter $\delta(\theta_j) = (\theta_j - \hat{\theta}_j)/se(\hat{\theta}_j)$, where $se(\hat{\theta}_j)$ is the standard error of the estimate of the parameter θ_j, on the x-axis. Bates and Watts (1988) also incorporate a second set of axes on their profile t plots depicting the nominal confidence levels on the y-axis and θ_j values on the x-axis . In this way the nominal likelihood limits for θ_j can be read directly from the profile t plot as the points of intersection between the horizontal line corresponding to the nominal confidence level and the profile t plot (Bates and Watts, 1988, pp.206-207). If the model is linear the plot of $\tau(\theta_j)$ versus $\delta(\theta_j)$ is a straight line through the origin with unit slope and so the extent to which the profile t plot deviates from this straight line gives an indication of the nonlinearity associated with the particular parameter under investigation. It should be noted that Cook and Weisberg (1990) have introduced confidence curves which are essentially a variation on the profile t plots of Bates and Watts.

CHAPTER 3

The problem of finding a confidence interval for a nonlinear function of the parameters, $g(\boldsymbol{\theta})$, using the profile likelihood method is not straightforward (Cook and Weisberg, 1990). The procedure amounts to finding estimates by maximising the likelihood function, or equivalently, in the case of the normal distribution, by minimising the error sum of squares, subject to a nonlinear constraint i.e. by maximising $l(\boldsymbol{\theta}|\mathrm{x})$ or minimising $S(\boldsymbol{\theta})$ for fixed $g(\boldsymbol{\theta})$. This problem has been tackled by very few authors. Clarke (1987) and Vecchia and Cooley (1987) offer approximations to the profile likelihood method for determining confidence intervals for a nonlinear function of the parameters, $g(\boldsymbol{\theta})$, based on finding extreme values of $g(\boldsymbol{\theta})$ over a joint confidence region for $\boldsymbol{\theta}$. Clarke and Grau (1995) propose a method for calculating profile likelihoods of a function of the parameters of a regression model and of a generalised linear model in which an artificial datum point is added to the sample and the change in the log likelihood due to this addition is used to create the profile likelihood function. These techniques are not easy to implement however.

In certain cases it is possible to transform the model so that the nonlinear function under consideration appears as a parameter in the model. Then the method described for an individual parameter of a model can be used to determine confidence limits for the nonlinear function of interest. For the models examined in the present study, the function $\eta(x, \boldsymbol{\theta})$ can be transformed so that the predicted response $\eta(x_g, \boldsymbol{\theta})$, where x_g is a given value of x, appears as a parameter in the model. This technique

is particularly effective if at least one of the parameters, say θ_1, occurs linearly in the model and the expected response $\eta(x, \boldsymbol{\theta})$ has the form

$$\eta(x, \boldsymbol{\theta}) = \theta_1 + \eta_1(x, \boldsymbol{\theta}_{(-1)}) \tag{3.9}$$

or the multiplicative form

$$\eta(x, \boldsymbol{\theta}) = \theta_1 \eta_2(x, \boldsymbol{\theta}_{(-1)}). \tag{3.10}$$

Hence in order to obtain confidence limits for $\eta_g = \eta(x_g, \boldsymbol{\theta})$, model (3.1) can be reparameterised as

$$\eta(x, \boldsymbol{\theta}) = \eta_g + \eta_1(x, \boldsymbol{\theta}_{(-1)}) - \eta_1(x_g, \boldsymbol{\theta}_{(-1)})$$

if $\eta(x, \boldsymbol{\theta})$ is of the form (3.9) or as

$$\eta(x, \boldsymbol{\theta}) = \eta_g \frac{\eta_2(x, \boldsymbol{\theta}_{(-1)})}{\eta_2(x_g, \boldsymbol{\theta}_{(-1)})}$$

if $\eta(x, \boldsymbol{\theta})$ is of the form (3.10). Thus η_g can now be regarded as a parameter in the model and is of course the parameter of interest. In the case of the nonlinear regression models describing an MLP and described in Chapter 2, this reparameterisation will always be possible provided bias terms are included in the network architecture. The profile log likelihood in the neighbourhood of $\eta(x_g, \hat{\boldsymbol{\theta}})$ can then be determined for η_g using the method described above for an individual parameter. The resultant plot of $S(\hat{\hat{\boldsymbol{\theta}}}_{(-1)} \mid \eta_g)$ against η_g is the profile log likelihood graph of the mean predicted value $\eta(x_g, \boldsymbol{\theta})$ and the $100(1-\alpha)\%$ confidence limits correspond to the two values of η_g that satisfy the equality in (3.8). This technique for constructing profile likelihoods

for nonlinear functions of the parameters is alluded to by Cook and Weisburg (1990) but not investigated further. This approach is thus introduced and investigated here.

The advantage of using likelihood over linearisation methods for constructing confidence intervals is that they are more theoretically sound and are able to capture features relating to the nonlinearity of the model such as the asymmetry in the distributions of the parameter estimates through asymmetrical confidence intervals. The linearisation method, on the other hand, assumes the nonlinearity is negligible and hence produces symmetric results. The disadvantage is however that such methods are substantially more computationally intensive.

3.4 Bootstrap Methods

The term bootstrap is thought to have originated from Rudolph Eric Raspe's eighteenth century, *Adventures of Baron Munchausen*, in which the Baron falls to the bottom of a deep lake and saves himself by pulling himself up by his own bootstraps. The bootstrap method was introduced in 1979 by Bradley Efron as an automatic, computer-based technique used to estimate the standard error of a parameter estimate, i.e. $se(\hat{\theta}_j)$, although the broad idea of resampling had been recognised for some time before then. The method itself is highly computer intensive meaning that it is very expensive in computer time but with the evolution of modern computing power this has become very much less of an issue.

CHAPTER 3

There are two general forms of bootstrapping, the parametric bootstrap and the nonparametric bootstrap. The underlying premise of the bootstrap method is that an observed data set $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is generated from an unknown probability distribution, denoted $f(\mathbf{x}, \boldsymbol{\theta})$. The difference between the two forms of bootstrapping is in how the estimation of $f(\mathbf{x}, \boldsymbol{\theta})$ takes place. When using the parametric bootstrap method $f(\mathbf{x}, \boldsymbol{\theta})$ is estimated by a parametric model, denoted $f(\mathbf{x}, \hat{\boldsymbol{\theta}})$, with the least squares or maximum likelihood estimate $\hat{\boldsymbol{\theta}}$ in place of the unknown parameters $\boldsymbol{\theta}$. For example $f(\mathbf{x}, \boldsymbol{\theta})$ may be assumed to be the normal distribution with mean μ and variance σ^2, then $f(\mathbf{x}, \hat{\boldsymbol{\theta}})$ corresponds to a normal distribution with mean $\bar{x}$ and variance s^2, where $\bar{x}$ and s^2 are the least squares estimates of μ and σ^2 respectively. The bootstrap procedure involves drawing B samples of size n from the distribution $f(\mathbf{x}, \hat{\boldsymbol{\theta}})$ by means of simulation, determining the parameter estimates for each such sample and then using these estimates to compile the distribution of $\hat{\boldsymbol{\theta}}$.

Nonparametric bootstrapping makes no assumption regarding the distribution of $f(\mathbf{x}, \boldsymbol{\theta})$ but rather relies on the empirical distribution function (e.d.f.) as an estimate of $f(\mathbf{x}, \boldsymbol{\theta})$. The e.d.f. is such that a probability of $\frac{1}{n}$ is associated with each observed value x_i, $i = 1, 2, \ldots, n$. As with the parametric method, B samples of size n are drawn from this discrete distribution, but the samples are drawn *with replacement* from the observations x_i, $i = 1, 2, \ldots, n$ to produce B new samples denoted by $\mathbf{x}^{*(b)} = (x_1^{*(b)}, x_2^{*(b)}, \ldots, x_n^{*(b)})$, $b = 1, \ldots, B$. Note that there are n^n possible bootstrap samples

of size n that can be drawn of which $n!$ of these do not contain any repetitions. As in the parametric case parameter estimates are established for each new data set and this information is in turn used to estimate the sampling distribution of $\hat{\theta}$. The parametric bootstrap is dependent on knowledge regarding the form of the underlying population distribution while the nonparametric bootstrap is less restrictive. For this reason only the nonparametric case is considered here.

3.4.1 Bootstrap Sampling

In the regression context, i.e. where the data is of the form (x_i, y_i), $i = 1, \ldots, n$, bootstrapping can take one of two forms; bootstrapping pairs or bootstrapping residuals. The procedures involved for each of these methods are described below.

Bootstrap Pairs Procedure: Consider a data set (x_i, y_i), $i = 1, \ldots, n$ with empirical distribution $\hat{f}(\mathbf{x}, \hat{\theta})$. The bootstrap pairs method consists of drawing a random sample of size n with replacement from the data pairs to generate a new data set denoted by (x_i^*, y_i^*), $i = 1, \ldots n$. This process is repeated B times to produce B new data sets $(x_i^{*(b)}, y_i^{*(b)})$, $i = 1, \ldots n$, $b = 1, \ldots, B$ (Efron and Tibshirani, 1993, p.78; Tibshirani, 1996).

CHAPTER 3

Bootstrap Residuals Procedure: A data set (x_i, y_i), $i = 1, \ldots, n$, is modelled appropriately using, for example, model (3.1) and an estimate, $\hat{\boldsymbol{\theta}}$, of the unknown parameters, $\boldsymbol{\theta}$, obtained. Then the normalised residuals are given by

$$e_i = [y_i - \eta(x_i, \hat{\boldsymbol{\theta}})]\sqrt{\frac{n}{n-p}} \quad i = 1, \ldots, n \tag{3.11}$$

(Wu, 1986). B random samples of size n are sampled with replacement from this set of residuals to give B sets of bootstrapped residuals denoted by $e_i^{*(b)}, i = 1, \ldots, n, b = 1, \ldots, B$. A new set of responses is then constructed by

$$y_i^{*(b)} = \eta(x_i, \hat{\boldsymbol{\theta}}) + e_i^{*(b)}$$

giving B new data sets denoted by $(x_i, y_i^{*(b)})$, $i = 1, \ldots n$, $b = 1, \ldots, B$ (Efron and Tibshirani, 1993, p.111; Tibshirani, 1996). Note that the explanatory variables are not bootstrapped, i.e. $x_i^{*(b)} = x_i, \quad i = 1, \ldots, n, \quad b = 1, \ldots, B$.

Bootstrapping of the residuals relies on the assumption that the model specified in (3.1) is correct and that the error terms are interchangeable. This is not always a valid assumption as is shown by means of an example in Efron and Tibshirani (1993, Section 9.5, pp. 113-114). The bootstrap pairs method is less sensitive to model assumptions, the only assumption being that the data pairs $(x_i,\ y_i)$ $i = 1, \ldots, n$ are randomly sampled from some distribution f, and is hence more robust than bootstrapping residuals.

3.4.2 Confidence Intervals

Efron and Tibshirani (1993) discuss various methods of constructing bootstrap confidence intervals for unknown parameters and some improvements to these methods which they claim give better coverage and stability. In this study only the percentile method and the extension of the percentile method to the BC_a method will be investigated and used. Since the bootstrap technique is an extremely computer intensive method a matter of concern is the appropriate number of bootstrap samples required for accurate inference while also maintaining computing efficiency. Accuracy is obtained through a large number of bootstrap samples but this necessitates an increase in computing time. Efron and Tibshirani (1993) investigated this dilemma quite thoroughly by studying the convergence of the function under investigation for a variety of B values and concluded that in the case of estimating the standard error of a parameter $B = 200$ should generally be sufficient (Efron and Tibshirani, 1993, p.52) whereas in the case of constructing confidence intervals $B = 1000$ is desirable (Efron and Tibshirani, 1993, Section 19.3, p.273).

The Percentile Method

Confidence intervals constructed using the percentile bootstrap method are based on the percentiles of the bootstrap distribution of the parameter of interest. The procedure is described in Efron and Tibshirani (1993, pp. 170-171) and is outlined in Box 3.1.

1. For each bootstrap sample generated by either the pairs or the residuals method calculate the least squares estimate, $\hat{\theta}_j^{*(b)}$, of the parameter of interest in the specified model.
2. Construct the empirical distribution, $\hat{G}$, of the bootstrap estimates, $\hat{\theta}_j^{*(b)}$, calculated in 1.
3. Find the percentiles, $\hat{G}^{-1}(\frac{\alpha}{2})$ and $\hat{G}^{-1}(1-\frac{\alpha}{2})$, which then form a $100(1-\alpha)\%$ confidence interval for the parameter θ_j.

Box 3.1 Percentile Confidence Interval Procedure for an Individual Parameter

The procedure described above can be extended to a nonlinear function of the model parameters and specifically to the predicted response $\eta(x_g, \boldsymbol{\theta})$. The amended procedure is outlined in Box 3.2.

A good confidence interval is one which is accurate in that it should give a coverage probability close to the nominal probability and correct in that the confidence limits should be relatively close to the exact confidence limits where these are known from statistical theory. The percentile interval has some desirable properties such as being able to pick up the shape of the distribution of the parameter of interest and being transform respecting, i.e. any transformation applied to the parameters can be directly applied to the confidence limits. However in practice these intervals tend to undercover, i.e. observed coverages are always less than or equal to the nominal

1. For each bootstrap sample generated by either the pairs or the residuals method calculate the least squares estimates, $\hat{\boldsymbol{\theta}}^{*(b)}$, of the parameters in the specified model.
2. Calculate the bootstrap predicted values, $\eta(x_g, \hat{\boldsymbol{\theta}}^{*(b)}), b = 1, \ldots, B$, for a range of x_g values.
3. Construct the empirical distribution, $\hat{G}$, of the bootstrap estimates, $\eta(x, \hat{\boldsymbol{\theta}}^{*(b)})$, calculated in 2.
4. Find the percentiles, $\hat{G}^{-1}(\frac{\alpha}{2})$ and $\hat{G}^{-1}(1 - \frac{\alpha}{2})$, which then form a $100(1-\alpha)\%$ confidence interval for the predicted value $\eta(x_g, \boldsymbol{\theta})$.

Box 3.2 Percentile Confidence Interval Procedure for a Predicted Response

coverage and in particular underestimate the tails of the distribution. Refinements to these intervals to correct for this bias were made resulting in the so-called bias-corrected and accelerated, abbreviated BC_a, confidence intervals and the approximate bootstrap, called ABC, confidence intervals (Efron and Tibshirani, 1993, p.178). It can be shown that the BC_a method is second-order accurate and second-order correct while the percentile interval is only first-order accurate and first-order correct (Efron and Tibshirani, 1993, pp. 321-326). Only the BC_a method is considered here.

CHAPTER 3

The BC_a Method

Confidence intervals calculated using the BC_a method depend on the two numbers $\hat{a}$ and $\hat{z}_0$, called the acceleration and bias-correction. The value of $\hat{a}$ is calculated in terms of the jackknife values of the statistic $\hat{\theta}_j$, where θ_j, $j = 1, \ldots p$, is the parameter of interest. Specifically for the set of training data with the ith point, (x_i, y_i), $i = 1, \ldots, n$, removed, let $\hat{\theta}_{j(i)}$ represents the parameter estimate calculated by omitting the ith point and define $\hat{\theta}_{j(.)} = \sum_{i=1}^{n} \hat{\theta}_{j(i)}/n$. The acceleration $\hat{a}$ is evaluated by

$$\hat{a} = \frac{\sum_{i=1}^{n}(\theta_{j(.)} - \hat{\theta}_{(i)})^3}{6\{\sum_{i=1}^{n}(\theta_{j(.)} - \hat{\theta}_{j(i)})^2\}^{3/2}} \tag{3.12}$$

(Efron and Tibshirani, 1993, pp. 185-188). A discussion and motivation of the derivation of $\hat{a}$ can be found in Efron (1987). The value of $\hat{z}_0$ is a measure of the difference between the bootstrapped parameter estimates, $\hat{\theta}_j^{*(b)}$, $b = 1, \ldots B$, and $\hat{\theta}_j$, or the median bias of $\hat{\theta}_j^{*(b)}, b = 1, \ldots, B$, evaluated on a normal scale. Specifically $\hat{z}_0$ is based on the number of bootstrap parameter estimates that are less than $\hat{\theta}_j$ and is given by

$$\hat{z}_0 = \Phi^{-1}\left(\frac{number\ of\ \hat{\theta}_j^{*(b)} < \hat{\theta}_j,\ b = 1, \ldots, B}{B}\right), \tag{3.13}$$

where $\Phi(.)$ represents the standard normal cumulative distribution function. If exactly half the bootstrap estimates are less than or equal to $\hat{\theta}_j$ then $\hat{z}_0 = 0$. The procedure then used to construct BC_a confidence limits for an unknown parameter θ_j is as in Box 3.1 but with step 3 replaced by the following scheme.

Find the percentiles, $\hat{G}^{-1}(\alpha_1)$ and $\hat{G}^{-1}(\alpha_2)$ with α_1 and α_2 given by

$$\alpha_1 = \Phi\left(\hat{z}_0 + \frac{\hat{z}_0 + z^{(\alpha/2)}}{1 - \hat{a}(\hat{z}_0 + z^{(\alpha/2)})}\right) \tag{3.14}$$

$$\alpha_2 = \Phi\left(\hat{z}_0 + \frac{\hat{z}_0 + z^{(1-\alpha/2)}}{1 - \hat{a}(\hat{z}_0 + z^{(1-\alpha/2)})}\right) \tag{3.15}$$

where $z^{(\alpha/2)}$ is the $100\frac{\alpha}{2}\,th$ percentile point of a standard normal distribution. $\hat{G}^{-1}(\alpha_1)$ and $\hat{G}^{-1}(\alpha_2)$ thus form a $100(1-\alpha)\%$ confidence interval for θ_j.

Note that when both $\hat{a}$ and $\hat{z}_0$ are equal to zero then $\alpha_1 = \frac{\alpha}{2}$ and $\alpha_2 = 1 - \frac{\alpha}{2}$ as in step 3 of Box 3.1.

The ideas presented here for an unknown parameter θ_j can be extended to find BC_a confidence limits for the predicted value $\eta_g = \eta(x_g, \boldsymbol{\theta})$. The value of $\hat{a}$ is calculated in terms of the jackknife values of the statistic $\eta(x, \hat{\boldsymbol{\theta}})$ with $\hat{\eta}_{(i)}$ representing the predicted value evaluated at x_g using the parameter estimates $\hat{\boldsymbol{\theta}}_{(i)}$ calculated by omitting the ith data point and given by $\hat{\eta}_{(.)} = \sum_{i=1}^{n} \hat{\eta}_{(i)}/n$. The acceleration $\hat{a}$ is therefore reformulated as

$$\hat{a} = \frac{\sum_{i=1}^{n}(\eta_{(.)} - \eta_{(i)})^3}{6\{\sum_{i=1}^{n}(\eta_{(.)} - \eta_{(i)})^2\}^{3/2}}. \tag{3.16}$$

The bias-correction, $\hat{z}_0$, is evaluated from the number of bootstrap predicted values

that are less than $\eta(x_g, \hat{\boldsymbol{\theta}})$ and is given by

$$\hat{z}_0 = \Phi^{-1}\left(\frac{number\,of\,\eta(x_g, \hat{\boldsymbol{\theta}}^{*(b)}) < \eta(x_g, \hat{\boldsymbol{\theta}}), b = 1, \ldots, B}{B}\right). \tag{3.17}$$

Step 4 of Box 3.2 is now replaced by the following procedure.

> Find the percentiles $\hat{G}^{-1}(\alpha_1)$ and $\hat{G}^{-1}(\alpha_2)$ with α_1 and α_2 given by (3.14) and (3.15) respectively which define a $100(1-\alpha)\%$ BC_a confidence interval for $\eta(x_g, \boldsymbol{\theta})$.

In Chapter 4 two specific examples are examined by applying the bootstrap percentile and BC_a methods using both the pairs and residuals methods and it is shown that for the specific model under investigation the bootstrap pairs method performs poorly in comparison to the bootstrap residuals.

Chapter 4

Applications and Results

4.1 Introduction

The methods discussed in Chapter 3 are now considered for two specific examples. The first example uses data describing (x, y) measurements made on bean root cells (Ratkowsky, 1983, p. 88) to which a single logistic model is fitted. This is analogous to an MLP comprising a single hidden-layer with one hidden unit with a logistic activation function and no bias term. The logistic model is known to be generally well-behaved in that it is close-to-linear and is hence used as a reference. The second example is a synthetic one in which the deterministic component of the model described in Section 3.1, is represented by the sum of two logistic functions and the requisite data is simulated. This example represents an MLP with a single hidden-layer with two hidden

units with logistic activation functions and a bias term. For each of these two examples standard errors and 95% confidence intervals for the predicted responses $\eta(x_g, \boldsymbol{\theta})$ for a set of given x values, x_g, were calculated using the methods described in Chapter 3, and compared by means of coverage probabilities.

A coverage probability is defined as the probability that a confidence interval with a nominal probability of $1-\alpha$ contains the true value of the parameter of interest. An observed coverage is the actual proportion of confidence intervals, constructed using a particular method, that contain the true parameter value. If the process producing the data is repeated a large number of times and the confidence intervals are exact then the observed coverages will approach the nominal coverages. If the confidence intervals are approximate, as in this study, the observed coverages will not approach $1-\alpha$ exactly but, if the approximation is reasonably good, the observed coverage should be close to the nominal value of $1-\alpha$. Hence comparison of observed and nominal coverage probabilities provides a useful tool for comparing different techniques for constructing confidence limits.

In the present study, observed coverage probabilities are obtained by simulating a large number of data sets from the true model, setting confidence limits to the parameter of interest, $\eta(x_g, \boldsymbol{\theta})$, and then forming the ratio of the number of these intervals that contain the true value to the number of simulations. For both of the examples considered in this chapter 500 data sets were simulated for specified values of

the parameters $\boldsymbol{\theta}$ and x, in line with the study of Donaldson and Schnabel (1987), and the coverage probabilities calculated as

$$\frac{\text{number of } \eta(x_g, \boldsymbol{\theta}) \in (\hat{L}_i, \hat{U}_i)}{500}, \quad i = 1, \ldots, 500, \tag{4.1}$$

where $\hat{L}_i$ and $\hat{U}_i$ denote the lower and upper confidence limits of a 95% confidence interval for the predicted response $\eta(x_g, \boldsymbol{\theta})$ for the *ith* data set respectively . The necessary programming was performed in GAUSS using the CURVEFIT module to fit the nonlinear models.

4.2 Bean Root Cells Example

The data consist of fifteen (x, y) pairs of measurements on bean root cells where x represents the distance from the tip of the root in intervals of 1 inch from 0.5 inches to 14.5 inches and y represents the water content in the bean root cell measured at the point x. The model fitted to the data is of the form

$$y_i = \frac{\theta_1}{1 + e^{-\theta_2 - \theta_3 x_i}} + \epsilon_i, \quad i = 1, \ldots, 15. \tag{4.2}$$

Before constructing confidence intervals for $\eta(x_g, \boldsymbol{\theta})$ the PE and IN curvature measures were evaluated to establish whether the model could be considered close-to-linear. As stated in Section 3.2 if the PE and IN measures are less than the cut-off value of $1/(2\sqrt{F}) = 0.268$, where $F = F_{3,12,0.05} = 3.49$, then the confidence intervals constructed

using linearisation and likelihood methods are expected to be close to exact and the coverage probabilitites should in turn be close to the nominal 95% level. The measures of curvature for the bean root data set are $PE = 0.372$ and $IN = 0.107$. While the intrinsic nonlinearity is less than the cut-off value of 0.268 the parameter effects curvature is not. However, on the basis of simulation studies, Ratkowsy (1983, pp.66-68) claims that the parameter effects curvature in this example is not serious and hence that the logistic model can be regarded as being close-to-linear.

4.2.1 Linearisation Method

To implement this method the error terms in (4.2) are assumed to be normally distributed with mean 0 and unknown variance σ^2. The least squares parameter estimates for $\boldsymbol{\theta}$ and σ^2 were found to be $\hat{\boldsymbol{\theta}} = (21.51, -3.957, 0.622)$ and $s^2 = 0.518$ respectively. Approximate 95% confidence intervals were calculated using (3.5) with $t^*_{12} = 2.179$ and are illustrated in Figure 4.1 as plots of $\pm t^*_{12} se[\eta(x_g, \hat{\boldsymbol{\theta}})]$ versus x_g, where $x_g \in [0.5, 14.5]$. Observed coverages were obtained by simulating 500 data sets for model (4.2) for selected values of $x_g = 1.5, 3.5, \ldots, 13.5$, with $\boldsymbol{\theta}$ and σ^2 equal to the least squares estimates $\hat{\boldsymbol{\theta}}$ and s^2 respectively, and these are presented in Table 4.1.

Method	x_g						
	1.5	3.5	5.5	7.5	9.5	11.5	13.5
Linearisation	0.954	0.958	0.962	0.962	0.946	0.942	0.948
Profile Likelihood	0.928	0.932	0.932	0.918	0.932	0.938	0.934
Percentile Bootstrap Pairs	0.910	0.902	0.908	0.886	0.894	0.890	0.886
Percentile Bootstrap Residuals	0.932	0.940	0.932	0.932	0.930	0.934	0.944
BCa Pairs	0.914	0.908	0.902	0.910	0.884	0.888	0.900
BCa Residuals	0.916	0.918	0.912	0.904	0.904	0.924	0.918

Table 4.1 : Coverage probabilities for the bean root data with a nominal level of 95% and 500 simulations.

4.2.2 Profile Likelihood Method

The profile likelihood method discussed in section 3.3 requires the calculation of conditional sums of squares $S(\widehat{\boldsymbol{\theta}} \mid \eta_g)$ for each value, x_g, which belongs to a chosen grid of x-values. These sums of squares are obtained by reparameterising the expected response (4.2) as

$$\eta(x, \boldsymbol{\theta}) = \eta_g \frac{(1 + e^{-\theta_2 - \theta_3 x_g})}{(1 + e^{-\theta_2 - \theta_3 x})},$$

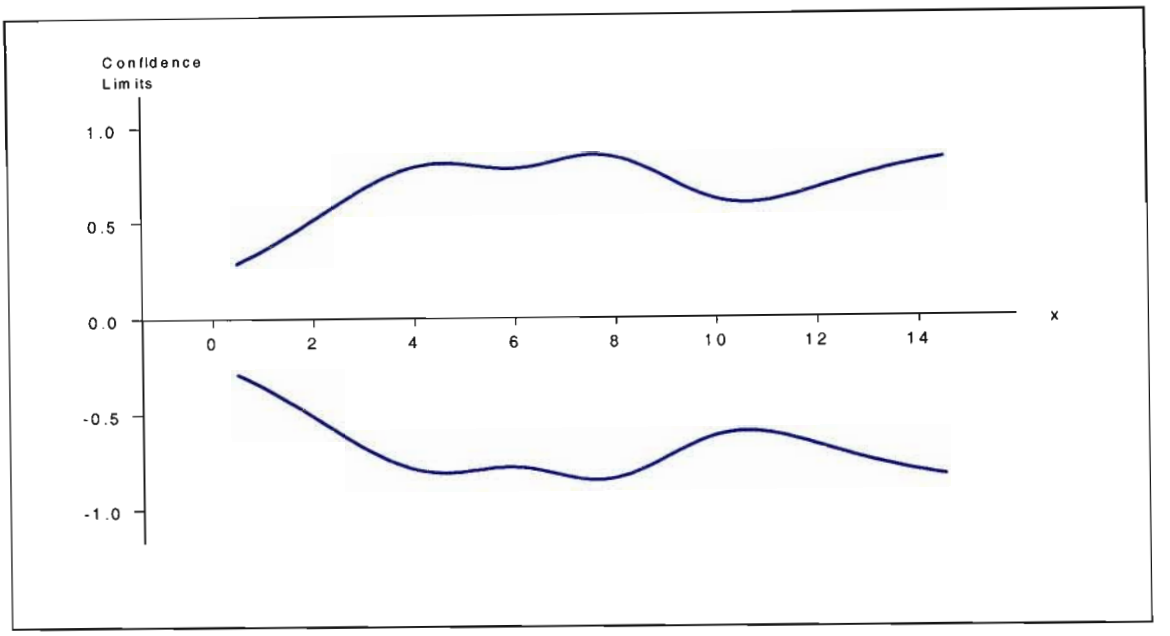

Figure 4.1: 95% Confidence limits for the predicted response for the bean root data using the linearisation method

where $\eta_g = \eta(x_g, \boldsymbol{\theta})$ is now a parameter in $\eta(x, \boldsymbol{\theta})$, and by invoking the GAUSS module CURVEFIT to minimise the associated error sum of squares, $S(\hat{\boldsymbol{\theta}} \mid \eta_g)$, with η_g fixed. For each x_g the resulting profile log-likelihood for η_g was found to be approximately quadratic, as illustrated in Figure 4.2 for $x_g = 5$, and the confidence limits corresponding to the two values of η_g satisfying the equality in (3.8) were readily obtained by means of the bisection method. A plot of the 95% confidence limits versus x_g for a fine grid of x_g values over the interval $[0.5, 14.5]$, together with the 95% confidence limits found using the linearisation method over the same x_g values, is shown in Figure 4.3.

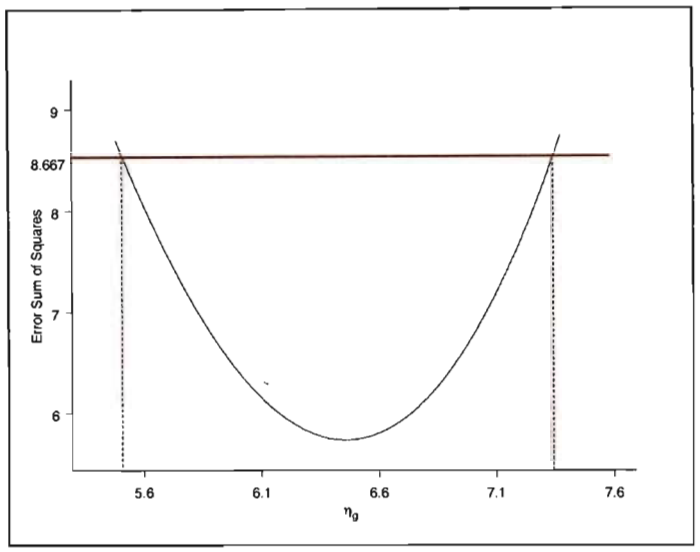

Figure 4.2: The profile likelihood graph for x_g=5

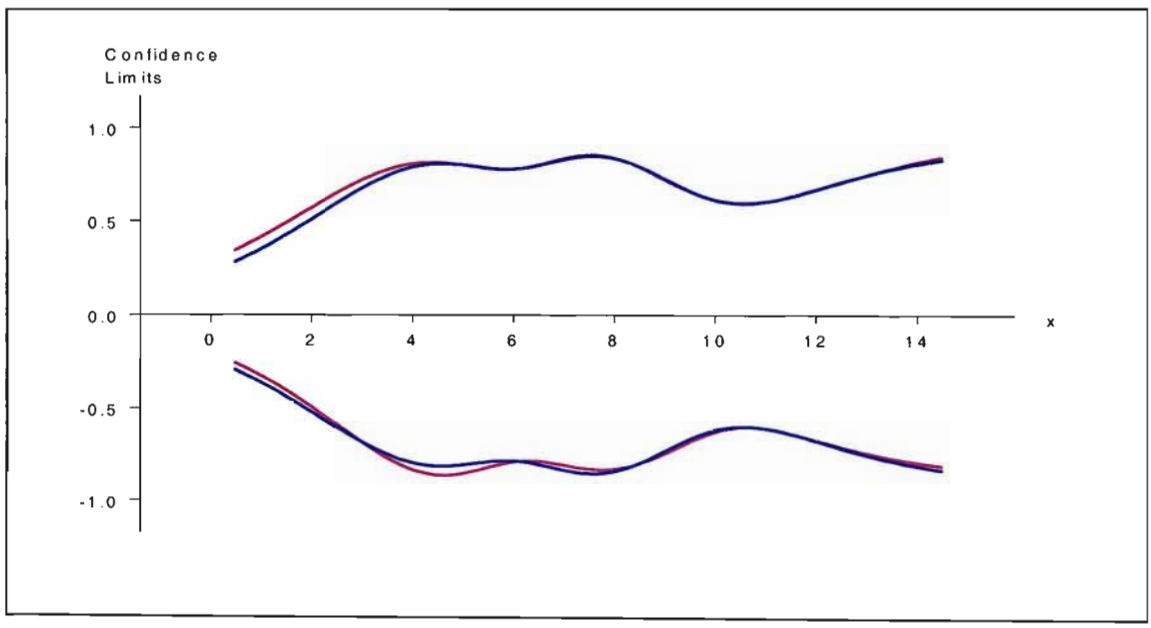

Figure 4.3: 95% Confidence limits for the predicted response for the bean root data using the profile likelihood method (pink) together with the linearisation method (blue)

Observed coverages for the profile likelihood intervals were calculated using the same 500 simulated data sets as for the calculation of coverages for the linearisation method and are given in Table 4.1.

4.2.3 Bootstrap Methods

4.2.3.1 Percentile Method

The procedure described in section 3.4.1 using both the bootstrap pairs and bootstrap residuals methods, was implemented for the bean root data. In each case 10000 bootstrap data samples $(x_i^{*(b)}, y_i^{*(b)})$, $i = 1, \ldots 15, b = 1, \ldots, 10000$, were generated from the original data and the least squares estimates $\hat{\boldsymbol{\theta}}^{*(b)}$ and the predicted responses, $\eta(x_g, \hat{\boldsymbol{\theta}}^{*(b)})$, for a fine grid of x_g values, $x_g \in [0.5, 14.5]$, obtained for each such data set. Approximate 95% confidence limits for $\eta(x_g, \boldsymbol{\theta})$ were determined by ordering the bootstrapped predicted responses in ascending order and selecting the $250th$ and the $9750th$ ordered predicted responses as the lower and upper confidence limits respectively. The 95% confidence limits obtained using the bootstrap pairs method together with the 95% linearisation confidence intervals from section 4.2.1 are illustrated in Figure 4.4 while the 95% confidence limits obtained by means of the bootstrap residuals method are depicted in Figure 4.5. The observed coverages were again determined from the 500 simulated data sets, as described in section 4.2.1, for both the bootstrap pairs and the bootstrap residuals methods and these are summarised in Table 4.1.

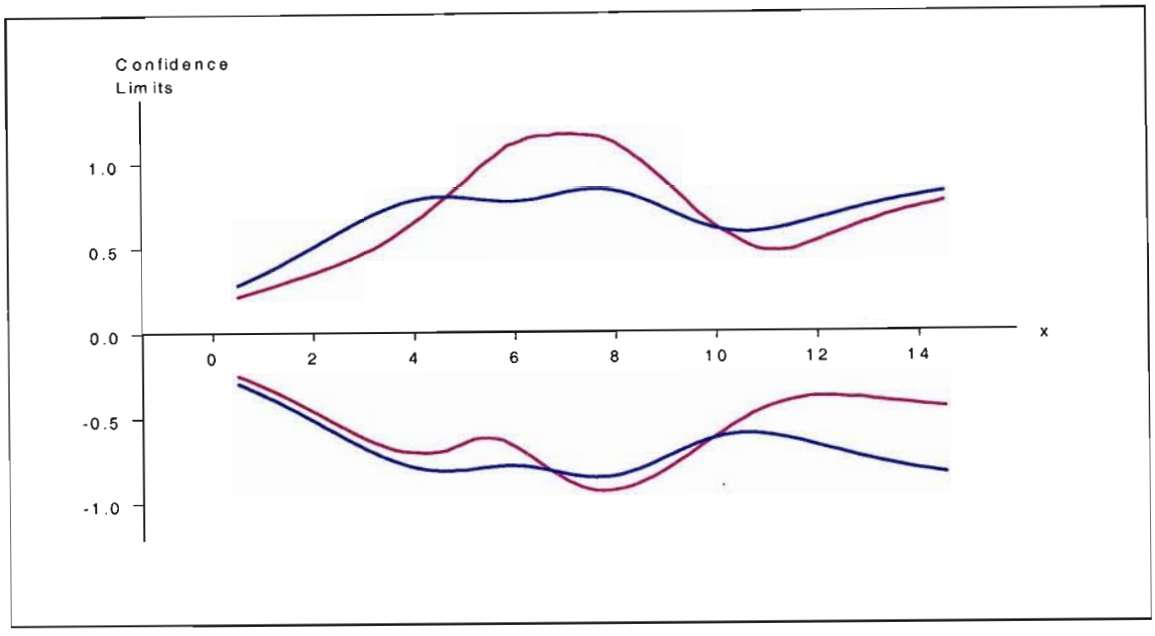

Figure 4.4: 95% Confidence limits for the predicted response for the bean root data using the percentile bootstrap pairs method (pink) together with the linearisation method (blue)

4.2.3.2 BC_a Method

BC_a confidence intervals were calculated for the true predicted response, $\eta(x_g, \boldsymbol{\theta})$, using the method described in section 3.4.2 for both the bootstrap pairs and residuals methods. Specifically 10000 bootstrap samples for each method were taken and the corresponding predicted responses, denoted by $\eta(x_g, \hat{\boldsymbol{\theta}}^{*(b)})$, $b = 1, \ldots 10000$, calculated and arranged in ascending order. The 95% confidence limits for $\eta(x_g, \boldsymbol{\theta})$ correspond to the $(100\alpha_1)th$ and the $(100\alpha_2)th$ percentiles of the distribution of the bootstrap pre-

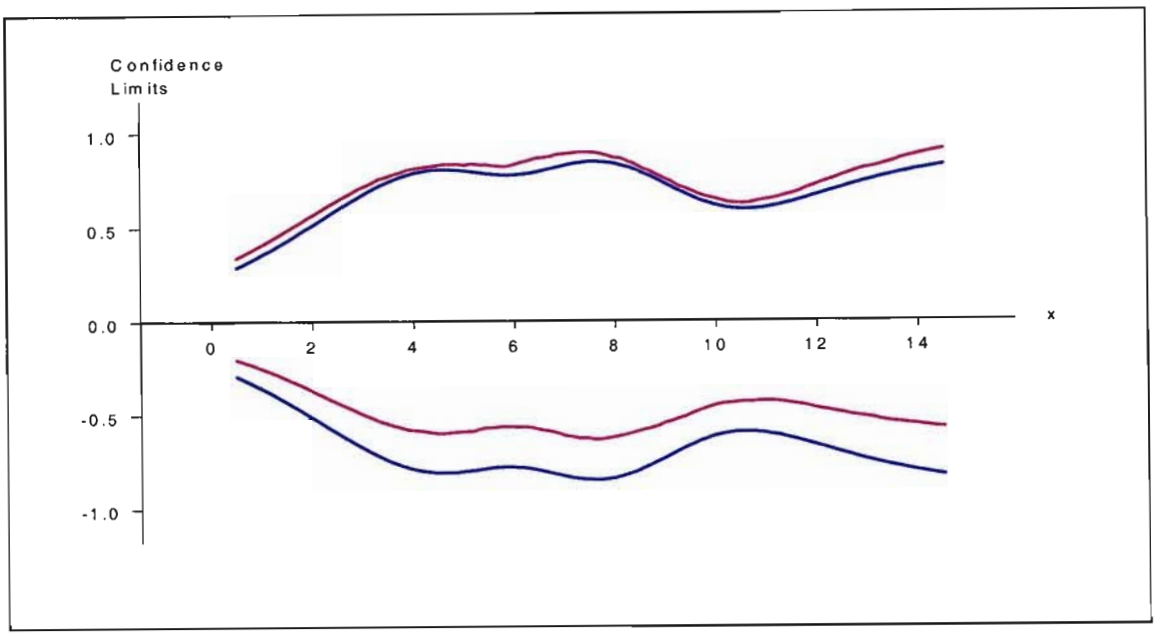

Figure 4.5: 95% Confidence limits for the predicted response for the bean root data using the percentile bootstrap residuals method (pink) together with the linearisation method (blue)

dicted responses, where α_1 and α_2 are determined by (3.14) and (3.15) respectively. The approximate 95% confidence limits obtained by means of the bootstrap pairs method together with the 95% confidence limits obtained by the linearisation method, both centred on $\eta(x_g, \hat{\boldsymbol{\theta}})$, are shown in Figure 4.6 while Figure 4.7 shows the approximate 95% bootstrap residual confidence limits together with the linearisation confidence limits. Coverages for each of the bootstrap methods were again determined using the 500 simulated data sets from section 4.2.1 and are recorded in Table 4.1.

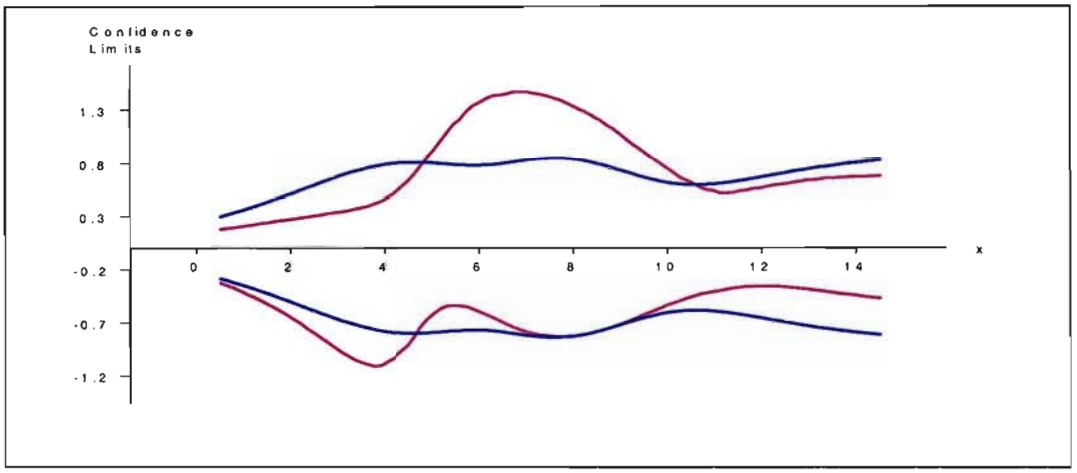

Figure 4.6: 95% Confidence limits for the predicted response for the bean root data using the BC_a pairs method (pink) together with the linearisation method (blue)

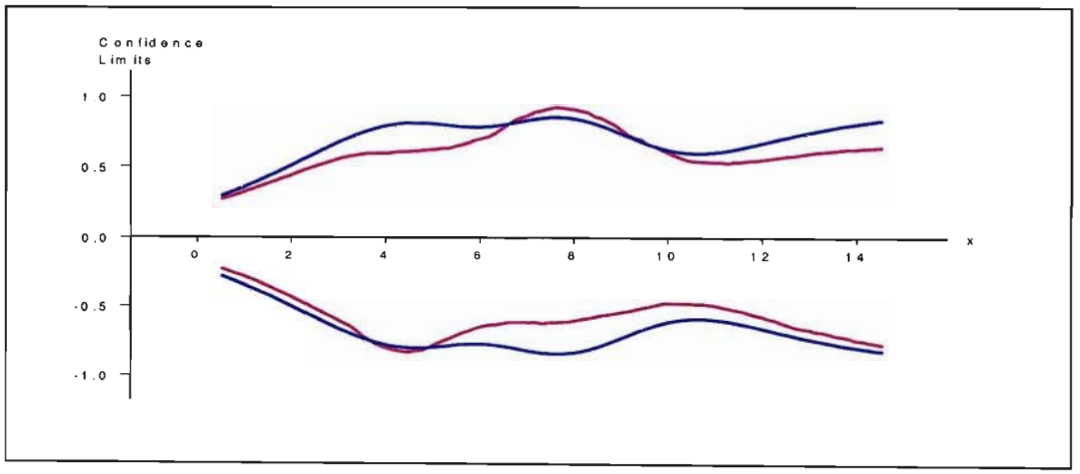

Figure 4.7: 95% Confidence limits for the predicted response for the bean root data using the BC_a residuals method (pink) together with the linearisation method (blue)

4.3 Sum of Two Logistics Example

The second example comprises an MLP with one hidden layer consisting of two nodes with logistic activation functions and a bias term and the associated nonlinear model is therefore of the form

$$y_i = \eta(x_i, \boldsymbol{\theta}) + \epsilon_i \quad i = 1, \ldots n \tag{4.3}$$

where

$$\eta(x, \boldsymbol{\theta}) = \theta_5 + \frac{\theta_6}{1 + e^{-\theta_1 - \theta_2 x}} + \frac{\theta_7}{1 + e^{-\theta_3 - \theta_4 x}}. \tag{4.4}$$

The data were generated from this model assuming normally distributed error terms with $\sigma = 0.01$. The parameter values were taken to be $\{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7\} = \{0.5, 0.5, 1, -1, 0.1, 1, 1.5\}$ and the x values were 25 equally spaced values in the interval $[-12, 12]$ which produced a function as depicted in Figure 2.7. The resultant generated y values together with the x values are presented in Table 4.2. Interest again focuses on the construction of 95% confidence intervals for the true predicted response, $\eta(x_g, \boldsymbol{\theta})$, where x_g is a value in the interval $[-12, 12]$, using each of the methods described in Chapter 3. As for the previous example the curvature measures were investigated before proceeding with the construction of the confidence intervals. The intrinsic nonlinearity and the parameter effects curvatures were calculated as $IN = 0.1411$ and $PE = 25.6250$. A curvature measure less than the cut-off value of $1/2\sqrt{2.58} = 0.311$ renders the model and data set under consideration close-to-linear. Thus the intrinsic nonlinearity

x	y	x	y
−12.0	1.5944	1.0	1.5749
−11.0	1.6226	2.0	1.3045
−10.0	1.6215	3.0	1.1668
−9.0	1.6114	4.0	1.1327
−8.0	1.6294	5.0	1.0776
−7.0	1.6492	6.0	1.0825
−6.0	1.6743	7.0	1.0827
−5.0	1.7077	8.0	1.0926
−4.0	1.7634	9.0	1.0846
−3.0	1.8317	10.0	1.1050
−2.0	1.8983	11.0	1.0972
−1.0	1.9395	12.0	1.0956
0.0	1.8281		

Table 4.2 : Data for the sum of two logistics example

curvature is less than the critical value of 0.311 but the parameter effects curvature is extremely large. According to Donaldson and Schnabel (1987) this should imply that confidence limits calculated by the linearisation method will have coverages far from the nominal value whereas confidence limits calculated by the likelihood method

should have coverages close to the nominal value. Again, the observed coverages for 95% confidence intervals obtained by each of the methods of Chapter 3 were calculated and used to compare the different techniques used to construct confidence intervals.

4.3.1 Linearisation Method

In order to use this particular technique normality of the error terms, i.e. $\epsilon_i \sim N(0, \sigma^2)$, $i = 1, \ldots, n$, must be assumed. In this particular example the data were generated such that this assumption is valid. The least squares parameter estimates for θ were found to be $\hat{\boldsymbol{\theta}}$ =(0.9926, 0.5745, 1.0562, −1.1189, 0.3498, 0.7446, 1.2593) and $s = 0.01$ was used as the estimate of the unknown standard deviation, σ. Confidence limits at the 95% level were then calculated for the true predicted response, $\eta(x_g, \boldsymbol{\theta})$, where x_g belongs to a fine grid of equally spaced values in the interval $[-12, 12]$ again using (3.5) with $t^* = 2.101$, and these are illustrated in Figure 4.8. Coverages were obtained by simulating 500 data sets from the model (4.3) using the true parameter values and are recorded in Table 4.3.for $x_g = \{-10.0, -7.5, \ldots, 10.0\}$.

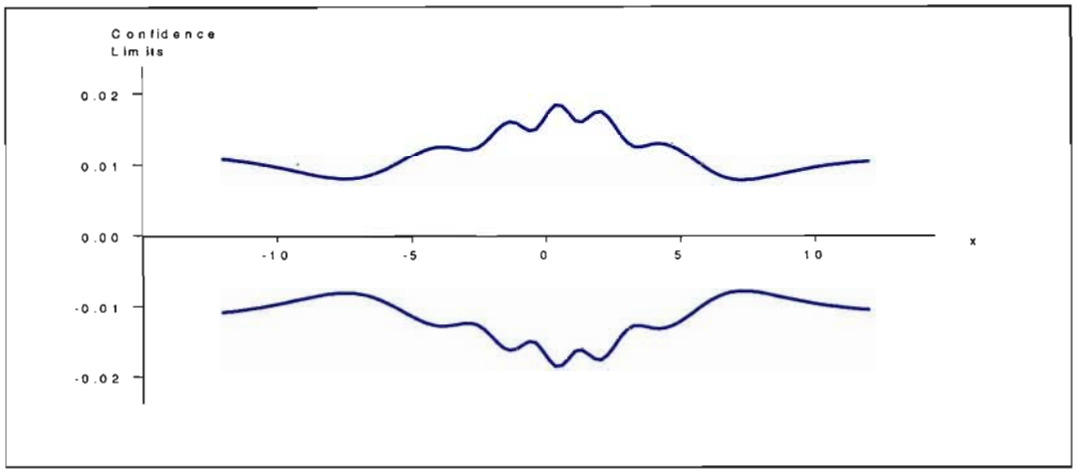

Figure 4.8: 95% Confidence limits for the predicted response for the sum of two logistics example using the linearisation method

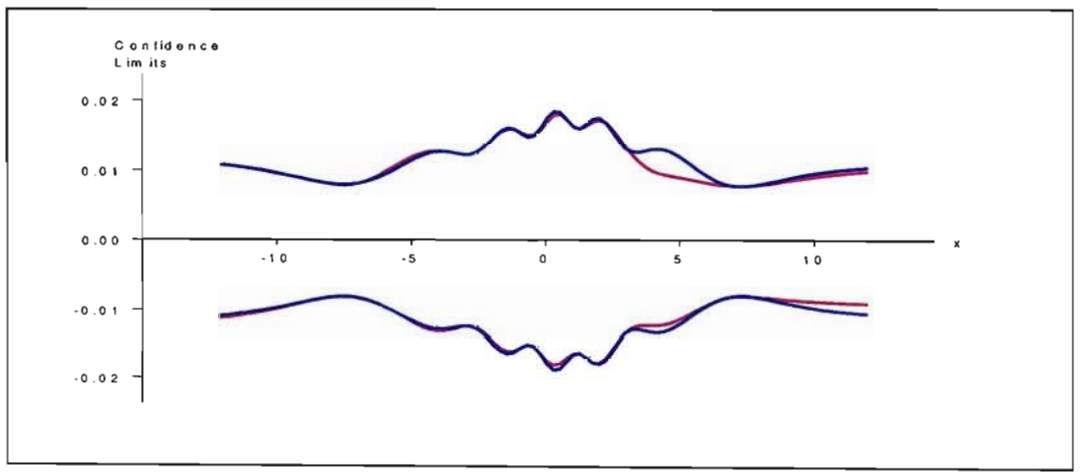

Figure 4.9: 95% Confidence limits for the predicted response for the sum of two logistics example using the profile likelihood method (pink) together with the linearisation method (blue)

Method	x_g								
	−10.0	−7.5	−5.0	−2.5	0	2.5	5.0	7.5	10.0
Linearisation	0.956	0.944	0.948	0.956	0.962	0.942	0.932	0.952	0.938
Profile Likelihood	0.932	0.938	0.948	0.948	0.958	0.954	0.956	0.940	0.952
Percentile Bootstrap Pairs	0.890	0.920	0.926	0.954	0.980	0.958	0.928	0.918	0.898
Percentile Bootstrap Residuals	0.922	0.942	0.928	0.936	0.920	0.934	0.926	0.920	0.914
BC_a Pairs	0.912	0.930	0.906	0.940	0.928	0.964	0.884	0.908	0.890
BC_a Residuals	0.922	0.938	0.920	0.912	0.940	0.924	0.914	0.928	0.936

Table 4.3 : Coverage probabilities for the sum of two logistics example with a nominal level of 95% and 500 simulations.

4.3.2 Profile Likelihood Method

The profile likelihood for $\eta(x_g, \boldsymbol{\theta})$ is obtained by reparameterising (4.4) as

$$\eta(x, \boldsymbol{\theta}) = \eta_g + \eta_1(x, \boldsymbol{\theta}) - \eta_1(x_g, \boldsymbol{\theta}) \tag{4.5}$$

where

$$\eta_1(x, \boldsymbol{\theta}) = \frac{\theta_6}{1 + e^{-\theta_1 - \theta_2 x}} + \frac{\theta_7}{1 + e^{-\theta_3 - \theta_4 x}}$$

and $\eta_g = \eta(x_g, \boldsymbol{\theta})$ is a parameter in (4.5) and finding the least squares estimates $\widehat{\widehat{\boldsymbol{\theta}}}_{(-5)}|\eta_g$ with corresponding conditional sum of squares $S(\widehat{\widehat{\boldsymbol{\theta}}}_{(-5)}|\eta_g)$ where x_g is chosen from a fine grid of x values, $x_g \in [-12, 12]$. The resulting profile log-likelihood was found to be approximately quadratic, the confidence limits were obtained as the solutions to the equality given by (3.8) with $t^* = 2.101$, and the approximate 95% confidence limits for $\eta(x_g, \boldsymbol{\theta})$ were again calculated through the bisection method for each of the given x values, x_g. The confidence intervals are illustrated together with the corresponding linearisation confidence limits in Figure 4.9 and the observed coverages calculated from the 500 simulated data sets used throughout this example are presented in Table 4.3.

4.3.3 Bootstrap Methods

4.3.3.1. Percentile Method

The data was sampled using both the bootstrap pairs and bootstrap residuals methods as described in section 3.4.1. For each method 10000 bootstrap samples were taken, denoted $(x_i^{*(b)}, y_i^{*(b)}), i = 1, \ldots, 25, b = 1, \ldots 10000$, and the corresponding predicted responses, $\eta(x_g, \hat{\boldsymbol{\theta}}^{*b)})$, calculated for the selected grid of x values. Following the procedure described in Box 3.2, the $\eta(x_g, \hat{\boldsymbol{\theta}}^{*b)})$ were placed in ascending order for each given x value, x_g, representing the distribution, $\hat{G}$, of the predicted responses, and the requisite 95% confidence limits are the $250th$ and the $9750th$ percentiles of $\hat{G}$. Figure 4.10 compares the 95% percentile bootstrap pairs confidence interval with the 95% linearisation

confidence interval obtained in section 4.3.1, and Figure 4.11 depicts the 95% percentile bootstrap residuals confidence interval together with the relevant linearisation confidence interval. The above process was repeated for each of the 500 simulated data sets from section 4.3.1 to determine the coverages of the percentile bootstrap pairs and the percentile bootstrap residuals confidence intervals and these coverages are presented in Table 4.3.

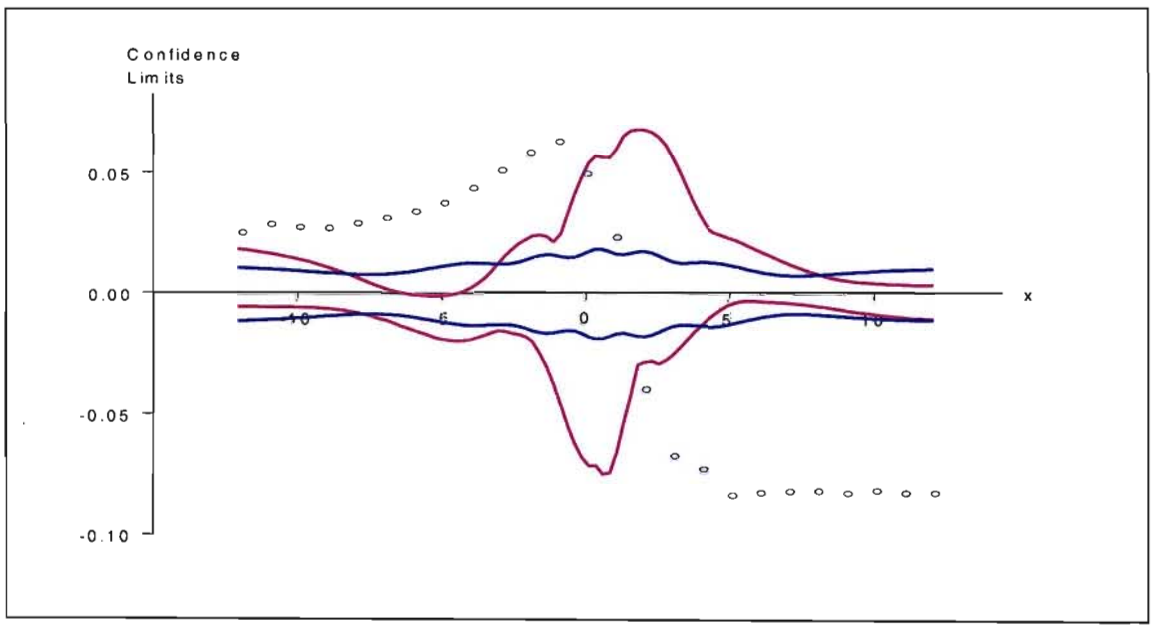

Figure 4.10: 95% Confidence limits for the predicted response for the sum of two logistics example using the percentile bootstrap pairs method (pink) together with the linearisation method (blue) and the scaled data points (circles)

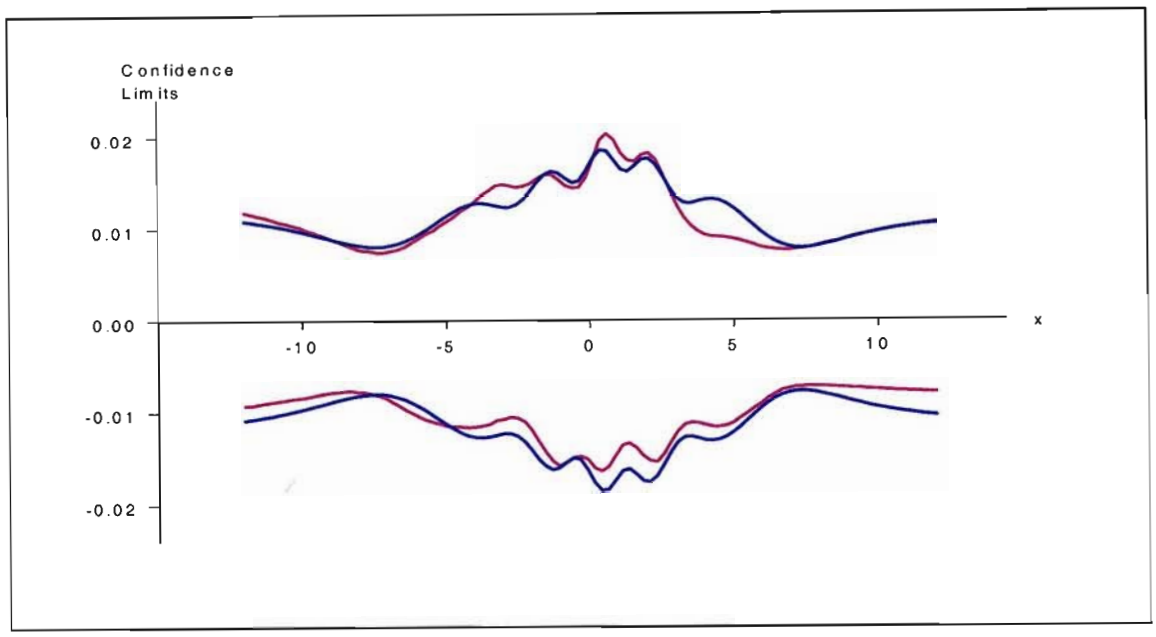

Figure 4.11: 95% Confidence limits for the predicted response for the sum of two logistics example using the percentile bootstrap residuals method (pink) together with the linearisation method (blue)

4.3.3.2. BC_a Method

The BC_a method was also implemented using both the bootstrap pairs and the bootstrap residuals methods. The data set was bootstrapped 10000 times and the bias, $\hat{z}_0$, and acceleration, $\hat{a}$, terms calculated according to (3.17) and (3.16) respectively, CURVEFIT was used to obtain the least squares estimates $\hat{\boldsymbol{\theta}}^{*(b)}$ and hence the predicted responses $\eta(x_g, \hat{\boldsymbol{\theta}}^{*(b)})$ which form the distribution $\hat{G}$. The 95% confidence limits for the predicted response $\eta(x_g, \boldsymbol{\theta})$ are the $(100\alpha_1)th$ and the $(100\alpha_2)th$ percentiles of

$\hat{G}$, α_1 and α_2 determined by (3.14) and (3.15) respectively. Figure 4.12 shows the 95% confidence limits obtained by means of the BC_a method using bootstrap pairs together with the 95% confidence limits obtained using the linearisation method. Figure 4.13 is essentially the same as Figure 4.12 but with the bootstrap residuals method used to calculate the BC_a confidence limits. The coverages were again obtained by means of the 500 simulated data sets and are presented in Table 4.3 separately for bootstrap pairs and bootstrap residuals methods.

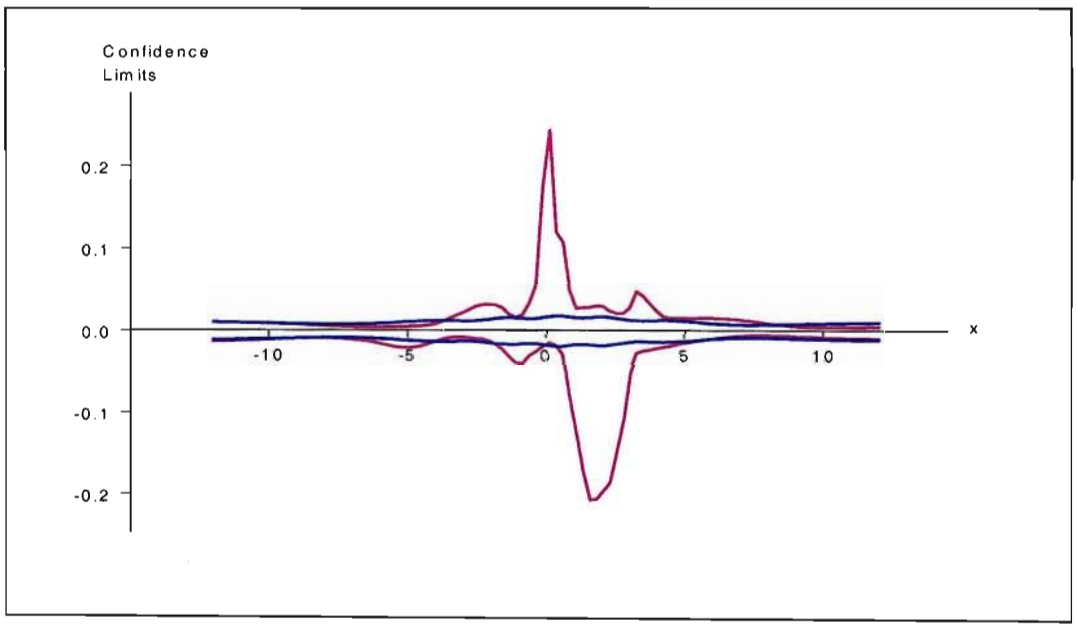

Figure 4.12: 95% Confidence limits for the predicted response for the sum of two logistics example using the BC_a pairs method (pink) together with the linearisation method (blue)

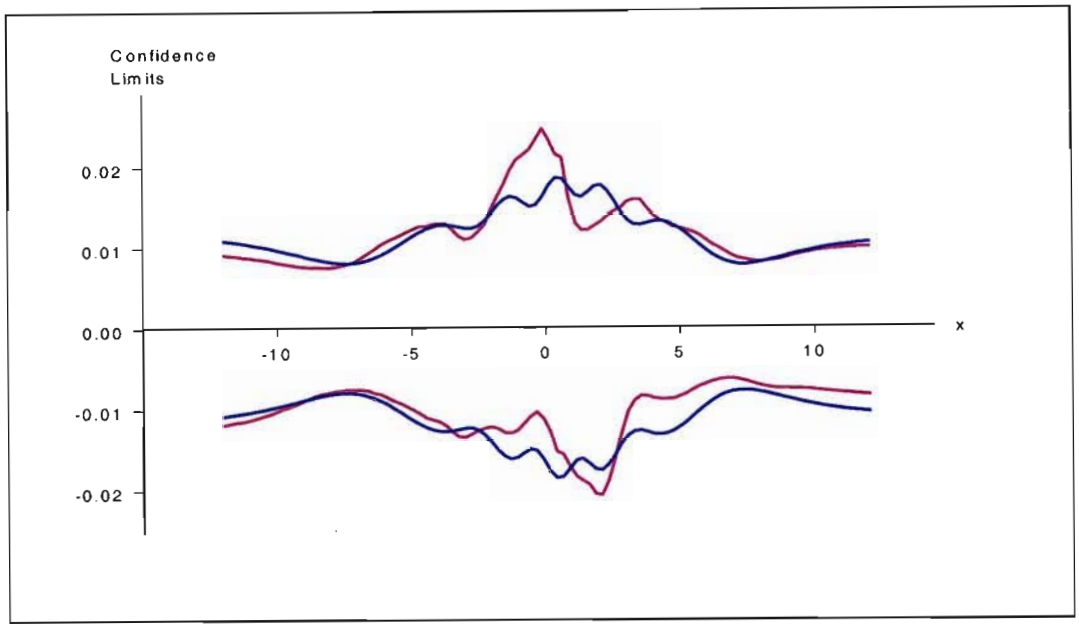

Figure 4.13: 95% Confidence limits for the predicted response for the sum of two logistics example using the BC_a residuals method (pink) together with the linearisation method (blue)

4.4 Comparison of Results

4.4.1 Bean Root Cell Example

As explained earlier the reason for selecting this particular example is the fact that the logistic model is known to be close-to-linear and should therefore behave similary to a linear model. The confidence limits obtained by the linearisation method and shown in Figure 4.1 were used as a reference for the other confidence interval techniques

considered in this study. Note that these confidence intervals are symmetric. The model is close-to-linear, confirmed by the PE and IN curvature measures, and thus the profile likelihood confidence intervals almost exactly mirror those of the linearisation method as illustrated in Figure 4.3. Table 4.1 contains the observed coverages obtained from 500 simulations of the bean root cell data for each of the techniques considered and it is clear that the linearisation method outperforms the other techniques, with coverages close to the nominal level of 95% over the specified grid of x values. The profile likelihood method coverages were not as close as expected to the nominal level of 95% and have a tendency to undercover over the domain of x values. The bootstrap techniques were however disappointing. Figure 4.4 illustrates the erratic nature of the bootstrap percentile pairs confidence intervals while Figure 4.5 depicts the bootstrap percentile residuals method where a distinct displacement at the lower limit is clearly evident. The coverages for the bootstrap percentile pairs and residuals methods, as given in Table 4.1, are very low, particularly in the case of the percentile pairs method. The BC_a confidence intervals did not improve the percentile confidence intervals. In fact the pairs method appeared to become even more erratic and this is illustrated in Figure 4.6. The BC_a residuals method corrected the displacement of the lower limit but these confidence limits were not as variable as the pairs method as illustrated in Figure 4.7. The observed coverages for the BC_a techniques, presented in Table 4.1, are again less than the nominal 95% level although perhaps not as severely in the case of

the percentile residuals. A general trait of the bootstrap confidence intervals appears to be that the observed coverages tend to be lower than the nominal 95% level over the entire domain of x values. This is consistent with the figures depicting the various confidence intervals.

4.4.2 Sum of Two Logistics Example

The observed coverages for the linearisation method, as detailed in Table 4.3, are good, with a slight undercovering on the upper tail. This is somewhat surprising in view of the fact that the parameter effects curvature measure is highly significant. Again the confidence limits, determined by the linearisation method and shown in Figure 4.8, were used for comparison with the other methods under consideration. The profile likelihood method also produced observed coverages close to the nominal 95% level and are presented in Table 4.3, but in contrast to the linearisation method exhibited a slight undercovering on the lower tail. This is seen quite clearly in Figure 4.9 with the profile likelihood confidence limits following closely the linearisation limits. Figures 4.10 and 4.11 depict the confidence intervals constucted using the percentile bootstrap pairs and residuals methods respectively and as with the bean root cell example the pairs method is erratic while the residuals method again follows the general shape of the linearistaion limits but with the lower limit systematically displaced. The coverages given in Table 4.3 reflect the poor performance of these limits, again with severe undercoverage over

the domain of x values. The BC_a method did not improve on the percentile method, and indeed the results from the BC_a method were extremely erratic and thus disappointing. The observed coverages for the BC_a methods are detailed in Table 4.3 and are generally far from the nominal 95% level and exhibit undercovering.

4.5 Summary

Overall the results for the models obtained by using both the linearisation and the profile likelihood methods were surpisingly good. In fact, to quote Wu (1986) "The linearisation method is a winner". In contrast the results obtained using the bootstrapping techniques were poor. The strength of the bootstrap methods is that they are based on the empirical distribution, $\hat{G}$, of the data and do not rely on linear or other approximations.

The poor performance of the bootstrap pairs method can, to some extent, be ascribed to the fact that there are only four data points defining the steep slope of the logistic model. This is shown in Figure 4.10 where the data points have been appropriately scaled and overlaid onto the percentile bootstrap pairs confidence limits. Indeed a straightforward calculation shows that

$$\text{P(at least one of the specified four is missing)} = \sum_{i=1}^{4} \binom{4}{i} \left(\frac{25-i}{25}\right)^{25} (-1)^{i+1} = 0.8463$$

and thus that the probability of omitting at least one of these points in a bootstrap

Chapter 5

Conclusion

In this study a statistical approach to hidden-layer feed forward neural networks or MLPs has been described and applied. The approach was found to be particularly powerful in that it allowed the use of statistical theory to develop confidence intervals for the predicted responses, which correspond to the outputs of a neural network. The methodology of developing such confidence intervals, which is relatively unexplored in the literature, is described and tested for two specific examples.

The results obtained for the three methods considered, i.e. the linearisation, profile likelihood and bootstrap methods, were interesting. The linearisation method gave good coverages and in addition is quick and easy to use. In comparison the profile likelihood method is a more sophisticated method than the linearisation method. An innovative and neat way of calculating the profile likelihood confidence intervals was

presented and implemented on the two specific examples considered in this study. In fact the confidence intervals obtained using the profile likelihood method produced very similar results to the linearisation method, but it is not at all evident that the results obtained are worth the additional computational effort in producing them.

The third method of constructing confidence intervals used in this study was the bootstrap method. It was thought that the bootstrap methods would perform well owing to the fact that bootstrapping is a nonparametric technique that relies on the empirical distribution of the data. Due to the nonlinear nature of the models under consideration, particularly in the case of the second example, the bootstrap methods were thus expected to give good coverages. The results obtained proved otherwise and the performance of the bootstrap methods was in fact disappointing. In particular the bootstrap pairs method performed very poorly but this can be attributed to poor sampling in that when resampling of the data takes place in order to form the bootstrap samples, there is a high probability of not sampling points that are crucial to the description of the function under consideration. The performance of the bootstrap residuals method was better in comparison to the bootstrap pairs method but could not compete with the likelihood-based methods. The BC_a method was implemented with the aim of improving the bootstrap confidence intervals, and while bias correction took place in the case of the bootstrap residuals method, the bootstrap pairs method produced even worse results, particularly in the case of the second example. The

bootstrap methods are highly computer intensive and time consuming and it is quite clear that there is no great benefit to be gained from using these techniques.

There are a number of interesting areas for future research emanating from the present study. The methods described were applied to small data sets and it would thus be interesting to apply the methods to large data sets. In addition some very interesting work on the Bayesian approach to neural networks has been produced (Ripley B. D., 1993; Bishop C. M., 1995; Neal, R. M., 1996; Ripley, B. D., 1996, pp. 163-168)and the Bayesian approach can be used to set confidence intervals to the predicted responses of these networks.

References

Aitken M. (1982). Direct Likelihood Inference. *GLIM82: Proceedings of the International Conference on Generalised Linear Models*, 76-86. Springer-Verlag, New York.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford.

Bates D. M. and Watts D.G. (1988). *Nonlinear Regression Analysis and Its Applications.* John Wiley and Sons, New York.

Brittain S. and Haines L.M. (1997). Nonlinear models for neural networks. *Mathematics of Neural Networks,* 8, 129-133.

Cheng B. and Titterington D. M. (1994) Neural Networks: A Review from a Statistical Perspective. *Statistical Science.*

Clarke G. P. Y. (1987). Approximate Confidence Limits for a Parameter Function in Nonlinear Regression. *Journal of the American Statistical Association*, 82, 221-230.

Clarke G. P. Y. and Grau E. A. (1995). A Useful Computational Method for Constructing Profile Likelihoods. Unpublished.

Cook R. D. and Weisberg S. (1990). Confidence Curves in Nonlinear Regression. *Journal of the American Statistical Association*, 85, 544-551.

Cox D. R. and Reid N. (1987). Parameter Orthogonality and Approximate Conditional Inference. *Journal of the Royal Statistical Society B*, 49, 1-39.

De Veaux R., Schumi J.,Scweinsberg J. and Ungar L. (1998). Prediction Intervals for

Neural Networks via Nonlinear Regression. *Technometrics,* 40, 273-282.

Donaldson J. R. and Schnabel R. B. (1987). Computational Experience with Confidence Regions and Confidence Intervals for Nonlinear Least Squares. *Technometrics*, 29, 67-82.

Efron B. (1979). Bootstrap Methods: Another look at the Jackknife. *Annals of Statistics*, 7, 1-26.

Efron B. (1987). Better Bootstrap Confidence Intervals. *Journal of the American Statistical Association*, 82, 171-200.

Efron B. and Tibshirani R. (1993). *An Introduction to the Bootstrap.* Chapman and Hall, New York.

Fine T. (1999). *Feedforward neural network methodology.* Springer, New York.

Hertz J., Krogh, A. and Palmer R. G. (1991). *Introduction to the Theory of Neural Computing.* Addison Wesley, Redwood City.

Hwang J.T.G. and Ding A.A. (1997). Prediction Intervals for Artificial Neural Networks. *Journal of the American Statistical Association,* 92, 748-757.

Geman S., Bienenstck E. and Doursat R. (1992). Neural Networks and the bias/variance dilemma. *Neural Computation,* 4, 1-58.

Nathan P. (1982). *The Nervous System.* Oxford University Press, Oxford.

Neal R. M. (1996). *Bayesian Learning for Neural Networks.* Springer, New York.

Press W. H., Flannery B. P., Teukolsky S. A. and Vetterling W. T. (1986). *Numerical*

recipes. The Art of Scientific Computing. Cambridge University Press, Cambridge.

Ratkowsky, D. A. (1983). *Nonlinear regression Modelling. A Unified Practical Approach.* Marcel Dekker, New York.

Ripley, B. D. (1993). *Statistical Aspects of Neural Networks. Networks and Chaos-Statistical and Probabilistic Aspects*, 40-123. Chapman and Hall, London.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks.* Cambridge University Press, Cambridge.

Ritter C. and Bates D.M. (1993) Profile Methods. Discussion Paper 9345. Center for Operations Research and Econometrics, Catholic University of Louvain.

Ritter C., Bisgaard S. and Bates D.M. (1994) A Comparison of Approaches to Inference for Nonlinear Models. *Computer Science and Statistics.* Proceedings of the 23^{rd} Symposium on the Interface, 148-155.

Seber, G. A. F. and Wild, C. J. (1989). *Nonlinear Regression.* John Wiley and Sons, New York.

Tibshirani R. (1996). A Comparison of Some Error Estimates for Neural Network Models. *Neural Computation*, 8, 152-163.

Vecchia A. V. and Cooley R. L. (1987). Simultaneous Confidence and Prediction Intervals for Nonlinear Regression Models with Application to a Groundwater Flow Model. *Water Resources Research*, 23, 1237-1250.

Wooldridge, D.E. (1963). *The Machinery of the Brain.* McGraw-Hill, New York.

Wu C. F. J. (1986). Jackknife, bootstrap and other resampling methods in regression analysis. *Annals of Statistics*, 14, 1261-1295.