A Statistical Approach to Neural Networks for Pattern Recognition

Robert A. Dunne
Commonwealth Scientific and Industrial Research Organization
Mathematical and Information Sciences
Statistical Bioinformatics-Health
North Ryde, New South Wales, Australia

BICENTENNIAL
1807
WILEY
2007
BICENTENNIAL

WILEY-INTERSCIENCE
A John Wiley & Sons, Inc., Publication

This Page Intentionally Left Blank

A Statistical Approach to Neural Networks for Pattern Recognition

BICENTENNIAL

1807
WILEY
2007

BICENTENNIAL

THE WILEY BICENTENNIAL–KNOWLEDGE FOR GENERATIONS

Each generation has its unique needs and aspirations. When Charles Wiley first opened his small printing shop in lower Manhattan in 1807, it was a generation of boundless potential searching for an identity. And we were there, helping to define a new American literary tradition. Over half a century later, in the midst of the Second Industrial Revolution, it was a generation focused on building the future. Once again, we were there, supplying the critical scientific, technical, and engineering knowledge that helped frame the world. Throughout the 20th Century, and into the new millennium, nations began to reach out beyond their own borders and a new international community was born. Wiley was there, expanding its operations around the world to enable a global exchange of ideas, opinions, and know-how.

For 200 years, Wiley has been an integral part of each generation's journey, enabling the flow of information and understanding necessary to meet their needs and fulfill their aspirations. Today, bold new technologies are changing the way we live and learn. Wiley will be there, providing you the must-have knowledge you need to imagine new worlds, new possibilities, and new opportunities.

Generations come and go, but you can always count on Wiley to provide you the knowledge you need, when and where you need it!

William J. Pesce

WILLIAM J. PESCE
PRESIDENT AND CHIEF EXECUTIVE OFFICER

Peter Booth Wiley

PETER BOOTH WILEY
CHAIRMAN OF THE BOARD

A Statistical Approach to Neural Networks for Pattern Recognition

Robert A. Dunne
Commonwealth Scientific and Industrial Research Organization
Mathematical and Information Sciences
Statistical Bioinformatics-Health
North Ryde, New South Wales, Australia

BICENTENNIAL
1807
WILEY
2007
BICENTENNIAL

WILEY-INTERSCIENCE
A John Wiley & Sons, Inc., Publication

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic format. For information about Wiley products, visit our web site at www.wiley.com.

Wiley Bicentennial Logo: Richard J. Pacifico

Library of Congress Cataloging-in-Publication Data is available.

ISBN 978-0-471-74108-4

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

To my father,
James Patrick

This Page Intentionally Left Blank

CONTENTS

NOTATION AND CODE EXAMPLES

g_n	$g_n \in \{1, \ldots, Q\}$, the class label, 9
Υ	matrix of weights in the MLP model, 11
Ω	matrix of weights in the MLP model, 11
$\mathrm{mlp}(x)$	the MLP fitted function, 11
$P.H.Q$	parameters describing an MLP model, 11
T	a target matrix, 11
t	a row of the target matrix T, 11
z^*	the MLP output, 11
ω	ω is an element in Ω, 11
υ	υ is an element in Υ, 11
R	$R = (P+1)H + (H+1)Q$ is the total number of weights, 12
ρ_l	sum of squares penalty function, 12
ρ_c	cross–entropy penalty function, 12
$U_X \Lambda_X V_X^T$	SVD of X, 21
$\mathbb{V}$	variance, 35
$\mathbb{E}$	expectation, 35
D_N	data set of size N, 37

P_e^A	the apparent error rate, 54
$\mathbb{D}$	deviance, 58
$\kappa(A)$	condition number, 62
$\xrightarrow{D}$	converges in distribution, 123
IC	influence curve, 124
SC	sensitivity curve, 144

There are a number of illustrative examples given at the end of some chapters. Many of these involve code in the `S` language, a high-level language for manipulating, analyzing and displaying data. There are two different implementations of `S` available, the commercial system *S-PLUS* (©Insightful Corporation, Seattle, WA `http://www.insightful.com`) and the open source `R` (R Development Core Team (2006), `http://www.R-project.org`). Invaluable references for the use of either system are Venables and Ripley (2000, 2002).

The two systems are implementations of the same `S` language (Becker et al., 1988), but are free to differ in features not defined in the language specification. The major differences that a user will see are in the handling of third party libraries and in the interfaces with C and Fortran code (Venables and Ripley (2000) give a summary of the differences).

We will use the `R` system in the examples, and will make use of the many third party contributed software libraries available through the "Comprehensive `R` Archive Network" (see `http://www.R-project.org`). Many of these examples will directly translate over to *S-PLUS* .

A number of the functions described here (and the code for the examples) are available through the website `http://www.bioinformatics.csiro.au/sannpr`. Many of the functions are packaged into the `mlp` `R` library.

PREFACE

This work grew out of two stimuli. One was a series of problems arising in remote sensing (the interpretation of multi-band satellite imagery) and the other was the lack of answers to some questions in the neural network literature.

Typically, in remote sensing, reflected light from the earth's surface is gathered by a sensing device and recorded on a pixel by pixel basis. The first problem is to produce images of the earth's surface. These can be realistic looking images but more likely (as they are more useful) they will be "false-colored" images where the coloring is designed to make some feature more visible. A typical example would be making vegetation covered ground stand out from bare ground.

However, the next level of problems in remote sensing concerns segmenting the image. Remote sensing gives rise to many problems in which it is important to assign a class membership label to the vector of pixel measurements. For example, each pixel in the image on page 90 has been assigned to one of the ground cover classes: "pasture"; "wheat"; "salt bush"; . . . , on the basis of its vector of measured light reflectances.

This was a problem that I worked on with the members of the Remote Sensing and Monitoring[1] project in Perth, Australia for many years during the 1990s as a PhD student, visiting academic and group member. The problems tackled by this group were (and are) of great practical importance. The group was heavily involved in developing methods of monitoring and predicting the spread of salinity in the farming areas of the south western corner of Australia. Due to changes in land use (clearing and broadacre farming) many areas of the world are experiencing similar problems with spreading salinity.

[1] http://www.cmis.csiro.au/RSM

To monitor salinity the group utilized the resources of the archived data collected by the LandSat series of satellites (jointly managed by NASA and the U.S. Geological Survey) with information going back to the early 1970s. The approach of the Remote Sensing and Monitoring group was unique in that, due to the large area it monitors and the need to gain an historic perspective on the salinity process, it pursued the use of lower resolution and more limited bandwidth data. This is different to many methodology groups in remote sensing who are keen to explore data from the latest high resolution multi-band sensors. It meant that a major focus of the group was on quite difficult classification problems.

It was in this context that I started looking at neural networks. While I found numerous explanations of how "multi-layered perceptrons" (MLPs) worked, time and time again I found, after working through some aspect of MLPs: implementing it; thinking about it; that it was a statistical concept under a different terminology.

This book then is partly the result of bringing the MLP within the area of statistics. However, not all statisticians are the same. There is a decided emphasis on robustness in this book and very little discussion of Bayesian fitting methods. We are all shaped by the times we lived through and by the questions we have struggled with. I have always worked with data sets that are large either in terms of observations (remote sensing) or variables (bioinformatics). It seems to me that training times for MLP models are long enough already without introducing Bayesian procedures. However there will always be a need to ensure that the model is resistant to anomalies in the data.

How to read this book

Chapters 1–5 describe the MLP model and show how it relates to some other statistical models used for classification tasks, such as linear discriminant analysis and logistic regression. This could form the basis for a self contained graduate level course on MLP models.

Chapters 6 and 7 describe adaptions of the MLP model to situations with large numbers of classes and to some image problems.

chapter	title
1	The Perceptron
2	The Multi–Layer Perceptron Model
3	Linear Discriminant Analysis
4	Activation and Penalty Functions
5	Model Fitting and Evaluation
6	The Task–Based MLP
7	Incorporating Spatial Information into an MLP Classifier

Chapters 8 and 9 investigate the robustness of the MLP model. The reader who is not interested in all the detail of Chapters 8 and 9 could read the summaries on pages 139 and 157 respectively.

Chapters 10 to 13 describe extensions and modifications to the MLP model. Chapter 10 describes a fitting procedure for making the MLP model more robust. Chapter 11 describes a modification for dealing with spectral data. Chapter 12 and 13 further investigate the modification of the weights during the fitting procedure and suggest some extensions to the MLP model based on this.

The book should give a largely self contained treatment of these topics but relies on at least an undergraduate knowledge of statistics and mathematics.

chapter	title
8	Influence Curves for the Multi-layer Perceptron Classifier
9	The Sensitivity Curves of The MLP Classifier
10	A Robust Fitting Procedure for MLP Models
11	Smoothed Weights
12	Translation Invariance
13	Fixed-slope Training

ROB DUNNE

Sydney, Australia

This Page Intentionally Left Blank

ACKNOWLEDGMENTS

This work owes a lot to Dr Norm Campbell, of the CSIRO Mathematics for Mapping & Monitoring group. Many of the approaches and ideas herein arose from discussions with Norm over the years.

Other people I would like to thank are Dr Harri Kiiveri of the CSIRO, for many helpful discussions and many insights and Professor Ian James, of Murdoch University. I would like also to thank Dr Ron McKay (formerly Vice-Chancellor of the Northern Territory University). Their openness to new ideas and ability to provide insightful criticism were what made this work possible.

The Martin's Farm data set (considered in Section 6.4 and other places) was made available by the Australian Centre for Remote Sensing (ACRES), a division of Geoscience Australia `http://www.ga.gov.au`.

Roger Clark of the United States Geological Survey also provided data and analysis suggestions.

R. A. D.

This Page Intentionally Left Blank

CHAPTER 1

INTRODUCTION

Neural networks were originally motivated by an interest in modelling the organic brain (McCulloch and Pitts, 1943; Hebb, 1949). They consist of independent processors that return a very simple function of their total input. In turn, their outputs form the inputs to other processing units. "Connectionism" was an early and revealing name for this work as the capabilities of the brain were felt to lie in the connections of neurons rather than in the capabilities of the individual neurons. Despite many debates over the years about the biological plausibility of various models, this is still the prevailing paradigm in neuro-science.

Important sources for the history of "connectionism" are McCulloch and Pitts (1943), Hebb (1949), Rosenblatt (1962), Minsky and Papert (1969), and Rumelhart et al. (1986). Anderson and Rosenfeld (1988) reproduce many of the historic papers in one volume and Widrow and Lehr (1990) give a history of the development.

However, the modern area of neural networks has fragmented somewhat and there is no attempt at biological plausibility in the artificial neural networks that are used for such tasks as grading olive oil (Goodacre et al., 1992), interpreting sonar signals (Gorman and Sejnowski, 1988) or inferring surface temperatures and water vapor content from remotely sensed data (Aires et al., 2004).

This book, and artificial neural networks in general, sit somewhere in a shared space between the disciplines of Statistics and Machine Learning (ML), which is in turn a cognate discipline of Artificial Intelligence. Table 1.1 summarizes some of the correspondences between the discipline concerns of Machine Learning and Statistics.

Table 1.1 Some correspondences between the discipline concerns of Machine Learning and Statistics.

machine learning		statistics
supervised learning	"learning with a teacher"	classification
unsupervised learning	"learning without a teacher"	clustering

Friedman (1991b) carries the teacher analogy further and suggests that a useful distinction between ML and statistics is that statistics takes into account the fact that the "teacher makes mistakes." This is a very accurate and useful comment. A major strand of this work is understanding what happens when "the teacher makes a mistake" and allowing for it (this comes under the heading of "robustness" in statistics).

Clearly one of the differences between ML and statistics is the terminology, which of course is a function of the history of the disciplines. This is exacerbated by the fact that in some cases statistics has its own terminology that differs from the one standard in mathematics. Another easily spotted difference is that ML has better names for its activities[1].

More substantial differences are that:

- machine learning tends to have a emphasis on simple, fast heuristics. It has this aspect in common with data mining and artificial intelligence;
- following on from the first point, whereas statistics tends to start with a model for the data, often there is no real data model (or only a trivial one) in machine learning.

Breiman in his article "Statistical modelling: The two cultures" (2001) talks about the divergence in practice between Statistics and Machine Learning and their quite different philosophies. Statistics is a "data modeling culture," where a function $f : x \rightarrow y$ is modeled in the presence of noise. Both linear and generalized linear models fall within this culture. However, Machine Learning is termed by Breiman an "algorithmic modeling culture." Within this culture, the function f is considered both unknown and unknowable. The aim is simply to predict a y value from a given x value.

Breiman argues that in recent years the most exciting developments have come from the ML community rather than the statistical community. Among these developments one could include:

- decision trees (Morgan and Sonquist, 1963);
- neural networks, in particular perceptron and multi-layer perceptron (MLP) models;
- support vector machines (and statistical learning theory) (Vapnik, 1995; Burges, 1998);
- boosting (Freund and Schapire, 1996).

[1] I believe that this comment was made in the lecture for which Friedman (1991b) are the notes.

Breiman in what, from the discussion[2], appears to have been received as quite a provocative paper, cautions against ignoring this work and the problem areas that gave rise to it.

While agreeing with Breiman about the significance of working with the algorithmic modeling culture we would make one point in favor of the statistical discipline. While these methodologies have been developed within the machine learning community, the contribution of the statistical community to a full understanding of these methodologies has been paramount. For example:

- decision trees were put on a firm foundation by Breiman et al. (1984). They clarified the desirability of growing a large tree and then pruning it as well as a number of other questions concerning the splitting criteria;

- neural networks were largely clarified by Cheng and Titterington (1994); Krzanowski and Marriott (1994); Bishop (1995a) and Ripley (1996) amongst others. The exaggerated claims made for MLP models prior to the intervention of statisticians no longer appear in the literature;

- boosting was demystified by Friedman et al. (1998) and Friedman (1999, 2000);

- support vector machines have yet to be widely investigated by statisticians, although Breiman (2001) and Hastie et al. (2001) have done a lot to explain the workings of the algorithm.

Brad Efron, in discussing Breiman's paper, suggests that it appears to be an argument for "black boxes with lots of knobs to twiddle." This is a common statistical criticism of ML. It arises, not so much from the number of knobs on the black box, which is often comparable to the number of knobs in a statistical model[3], but from the lack of a data model.

When we have a data model, it gives confidence in the statistical work. The data model arises from an understanding of the process generating the data and in turn assures us that we have done the job when the model is fitted to the data. There are then generally some diagnostic procedures that can be applied, such as examining the residuals. The expectation is that, as the data model matches the process generating the data, the residuals left over after fitting the model will be random with an appropriate distribution. Should these prove to have a pattern, then the modelling exercise may be said to have failed (or to be as good as we can do). In the absence of a data model, there seems little to prevent the process being reduced to an ad-hock empiricism with no termination ctiteria.

However the ML community is frequently working in areas where no plausible data model suggests itself, due to our lack of knowledge of the generating mechanisms.

[2] As Breiman (2001) was the leading paper in that issue of *Statistical Science*, it was published with comments from several eminent statisticians and a rejoinder from Leo Breiman.

[3] In some instances the number of "knobs" may be fewer for ML algorithms than for comparable statistical models. See Breiman's rejoinder to the discussion.

1.1 THE PERCEPTRON

We will start with Rosenblatt's perceptron learning algorithm as the foundation of the area of neural networks. Consider a set of data as shown in Figure 1.1. This is a 2-dimensional data set in that 2 variables, x_1 and x_2, have been measured for each observation. Each observation is a member of one of two mutually exclusive classes labelled "×" and "+" in the figure.

To apply the perceptron algorithm we need to have a numeric code for each of the classes. We use

$$y = \begin{cases} 1 & \text{for class } \times \text{ and} \\ -1 & \text{for class } + . \end{cases}$$

The model then consists of a function f, called an "activation function," such that:

$$f = \begin{cases} 1 & \omega_0 + \omega^T x \geq 0 \\ -1 & \text{otherwise.} \end{cases}$$

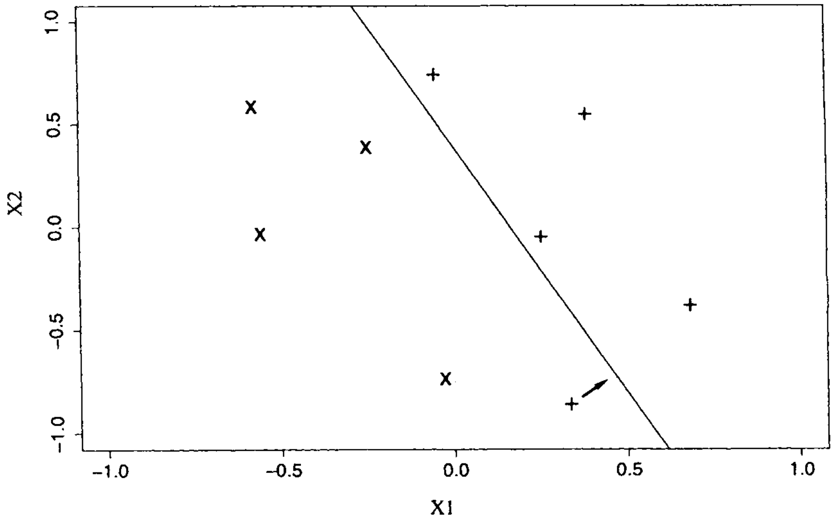

Figure 1.1 *A 2-dimensional data set consisting of points in two classes, labelled "×" and "+". A perceptron decision boundary $\omega_0 + \omega_1 x_1 + \omega_2 x_2$ is also shown. One point is misclassified, that is, it is on the wrong side of the decision boundary. The margin of its misclassification is indicated by an arrow. On the next iteration of the perceptron fitting algorithm (1.1) the decision boundary will move to correct the classification of that point. Whether it changes the classification in one iteration depends on the value of η, the step size parameter.*

The perceptron learning algorithm tries to minimize the distance of a misclassified point to the straight line $\omega_0 + \omega_1 x_1 + \omega_2 x_2$. This line forms the "decision boundary" in that points are classified as belonging to class "×" or "+" depending

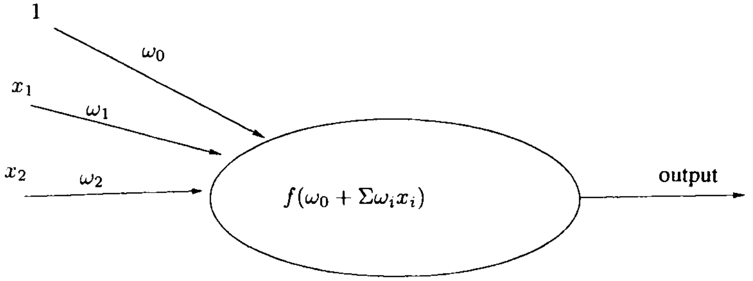

Figure 1.2 The perceptron learning model can be represented as a processing node that receives a number of inputs, forms their weighted sum, and gives an output that is a function of this sum.

on which side of the line they are on. The misclassification rate is the proportion of observations that are misclassified, so in Figure 1.1 it is 1/9. Two classes that can be separated with 0 misclassification error are termed "linearly separable."

The algorithm uses a cyclic procedure to adjust the estimates of the ω parameters. Each point x is visited in turn and the ωs are updated by

$$\omega_i \leftarrow \omega_i + \eta[y - f(\omega_0 + \omega^T x)]x. \tag{1.1}$$

This means that only incorrectly classified points move the decision boundary. The η term has to be set in advance and determines the step size. This is generally set to a small value in order to try to prevent overshooting the mark.

Where the classes are linearly separable it can be shown that the algorithm converges to a separating hyperplane in a finite number of steps. Where the data are not linearly separable, the algorithm will not converge and will eventually cycle through the same values. If the period of the cycle is large this may be hard to detect.

Where then is the connection with brains and neurons? It lies in the fact that the *algorithm* can be represented in the form shown in Figure 1.2 where a processing node (the "neuron") receives a number of weighted inputs, forms their sum, and gives an output that is a function of this sum.

Interest in the perceptron as a computational model flagged when Minsky and Papert (1969) showed that it was not capable of learning some simple functions. Consider two logical variables A and B that take values in the set {TRUE, FALSE}. Now consider the truth values of the logical functions AND, OR, and XOR (eXclusive OR, which is true if and only if one of its arguments is true) as shown in Table 1.2.

We can recast the problem of learning a logical function as a geometric problem by encoding {TRUE, FALSE} as $\{1, 0\}$. Now for the XOR function, in order to get a 0 classification error the perceptron would have to put the points $\{1, 1\}$ and $\{0, 0\}$ on one side of a line and $\{1, 0\}$ and $\{0, 1\}$ on the other. Clearly this is not possible (see Figure 1.3).

We note here the very different flavor of this work to traditional statistics. Linear discriminant analysis (see Chapter 3, p. 19, and references therein) is the classical statistical technique for classification. It can not achieve a zero error on the geo-

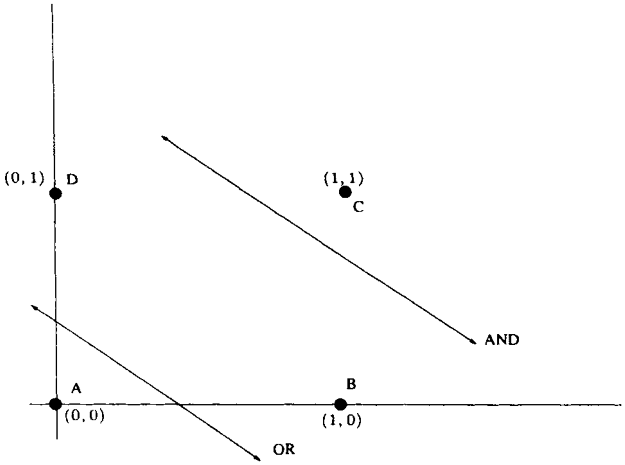

Figure 1.3 The problem of learning a logical function can be recast as a geometric problem by encoding {TRUE, FALSE} as {1, 0}. The figure shows decision boundaries that implement the function OR and AND. The XOR function would have to have points A and C on one side of the line and B and D on the other. It is clear that no single line can achieve that, although a set of lines defining a region or a non-linear boundary can achieve it.

metric XOR problem any more than the perceptron can. However, as far as I know, no statistician has ever shown a lot of concern about this fact.

Table 1.2 Consider two logical variables A and B that can take values in the set {TRUE, FALSE}. The truth values of the logical functions AND, OR, and XOR are shown.

A	B	AND	OR	XOR
T	T	T	T	F
F	F	F	F	F
T	F	F	T	T
F	T	F	T	T

Using a layered structure of perceptron as shown in Figure 2.1 (p. 10) overcame this problem and lead to a resurgence in interest in this research area. These are the "multi-layer perceptrons" (MLPs) that are the topic of this work. They required a different learning algorithm to the single perceptron and require that f be a differentiable function. It was the development of such algorithms that was the first step in their use. This has appeared several times in the literature, common early references being Werbos (1974) and Rumelhart et al. (1986).

Already a large number of questions are apparent: such as:

- what if there are more than two classes;
- what if the classes are not linearly separable – but there is a non-linear decision boundary that could separate them;

- do we want a classifier that performs well on this data set or on a new, as yet unseen, data set? Will they be the same thing?

These questions will be consider in the ensuing chapters.

David Hand has recently written a interesting paper entitled "Classifier Technology and the Illusion of Progress" (Hand, 2006). The progress that he is questioning is the progress of sophisticated techniques like MLPs, support vector machines (Vapnik, 1995; Burges, 1998), and others. He shows that for many examples, the decrease in classification error using sophisticated techniques is only marginal. It may be so small, in fact, that it may be wiped out by the vagaries and difficulties in attendance with real word data sets.

I think that David Hand's caution should be taken to heart. Sophisticated techniques should be used with caution and with an appreciation of their limitations and idiosyncrasies. If there is a good data model available, for example, an understanding that the data are Gaussian, then there may be no justification for using an MLP model.

MLP models have not always been used with an appreciation of their characteristics. The fact that MLPs can be used in a "black box" fashion, and seem to produce reasonable results without a lot of effort being put into modeling the problem, has often led to them being used in this way. It appears that MLPs were being used on hard problems, such as speech recognition and vision, long before any real groundwork was done on understanding the behavior of the MLP as a classifier[4].

This has led to debate in the literature on such elementary points as the capabilities of MLPs with one hidden layer[5], and a lack of understanding of the possible roles of hidden layer units in forming separating boundaries between classes. However, such understanding can be readily arrived at by considering the behavior of the MLP in simple settings that are amenable both to analytic and graphical procedures. In this book the simplest case of two classes and two variables is often used as an example and some points that have been debated in the literature may be amongst the first things that an investigator will notice when confronted with a graphical representation of the output function of an MLP in this simple setting.

The aim of this book is to reach a fuller understanding of the MLP model and *extend it* in a number of desirable ways. There are many introductions and surveys of multi-layer perceptrons in the literature (see below for references); however, none should be necessary in order to understand this book, which should contain the necessary introduction. Other works that could usefully be consulted to gain insight into the MLP model include Cheng and Titterington (1994), Krzanowski and Marriott (1994), Bishop (1995a), Ripley (1996) and Haykin (1999).

We use a number of examples from the area of remote sensing to illustrate various approaches. Richards and Jia (2006) is a good introduction to this problem area while Wilson (1992) and Kiiveri and Caccetta (1996) discuss some of the statistical issues involved. Once again, this work should be entirely self contained – with as much of the problem area introduced in each example as is needed for a full appreciation of the example. Multi-layer perceptrons have been used in the analysis of remotely sensed data in Bischof et al. (1992), Benediktsson et al. (1995)

[4] Early exceptions to this tendency to use MLPs without investigating their behavior are Gibson and Cowan (1990), Lee and Lippmann (1990) and Lui (1990). The situation has been changing markedly in recent years and many of the lacunae in the literature are now being filled.

[5] That is, are they capable of forming disjoint decision regions; see Lippmann (1987).

and Wilkinson et al. (1995). Paola and Schowergerdt (1995) give a review of the application of MLP models to remotely sensed data.