MEDICAL APPLICATIONS OF ARTIFICIAL NEURAL NETWORKS: CONNECTIONIST MODELS OF SURVIVAL

A DISSERTATION
SUBMITTED TO THE PROGRAM IN MEDICAL INFORMATION SCIENCES
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Lucila Ohno-Machado

March 1996

ABSTRACT

Although neural networks have been applied to medical problems in recent years, their applicability has been limited for a variety of reasons. One of those barriers has been the problem of recognizing rare categories. In this dissertation, I demonstrate, and prove the utility of, a new method for tackling this problem. In particular, I have developed a method that allows the recognition of rare categories with high sensitivity and specificity, and will show that it is practical and robust. This method involves the construction of sequential neural networks.

Rare categories occur and must be learned if practical application of neural-network technology is to be achieved. Survival analysis is one area in which this problem appears. In this work, I test the hypotheses that (1) sequential systems of neural networks produce results that are more accurate (in terms of calibration and resolution) than nonhierarchical neural networks; and (2) in certain circumstances, sequential neural networks produce more accurate estimates of survival time than Cox proportional hazards and logistic regression models. I use two sets of data to test the hypotheses: (1) a data set of HIV+ patients (AIDS Time-Oriented Health Outcome Study—ATHOS data set); and (2) a data set of patients followed prospectively for the development of cardiac conditions (Framingham data set).

Using the ATHOS data set, I show that a neural network model can predict death due to AIDS more accurately than a Cox proportional hazards model. Furthermore, I show that a sequential neural

network model is more accurate than a standard neural network model. Using the Framingham data set, I show that the predictions of logistic regression and neural networks are not significantly different, but that any of these models used sequentially is more accurate than its standard counterpart.

The sequential use of predictive models for survival analysis is advantageous because it makes better use of the available information. It often increases resolution with no sacrifice of calibration, as I demonstrate in this study. It also helps to delineate patterns of disease progression for individuals, rather than for groups of patients.

ACKNOWLEDGMENTS

It is hard to summarize in retrospect the important events that lead to the completion of this dissertation. When my husband Ruy and I arrived in the U.S. four and a half years ago, we carried suitcases loaded with books and minds loaded with dreams. Due to the warm and stimulating environment provided by friends and colleagues at Stanford, we were able to make some of these dreams come true. This dissertation is one of them, and it would not have been possible without the help and encouragement that I received at the Section on Medical Informatics at Stanford.

All members of my dissertation committee helped me focus my research and devise a reasonable schedule for each step of development and evaluation. Mark Musen, my principal adviser, has always supported my ideas and encouraged their pursuit. Mike Walker has helped me bring these ideas down to earth. I am grateful for his honest criticism and constructive advice (although sometimes it took me days to absorb it). Ted Shortliffe, whose example as a pioneer in combining the study of medicine and computer science has inspired me since my days of medical school in Brazil, helped me envision how this work would fit into the ever-growing field of medical informatics. Alan Garber patiently and critically assessed the quantitative aspects of this work, and made invaluable suggestions. Bill Brown and Daniel Bloch helped me with the evaluation methods. Dave Rumelhart was the person who introduced me to the field of neural networks, and whose patience I tested repeatedly every time I knocked at his door with a notebook full of questions and the curiosity and persistence of a two-year-old. I consider myself extremely lucky for having had the guidance of the scientist whose work on backpropagation has inspired research in so many domains.

My search for advice was not limited to my dissertation committee: Les Lenert, Russ Altman, and Yuval Shahar kindly donated their time for discussions and made multiple suggestions, which I hope to have adequately incorporated in this final work. Thank you for the technical and general advice concerning all aspects of this dissertation.

This work would have been possible (but extremely arduous!) without the technical support provided by Christopher Lane, Mike Macgirvin, and Jay Heyman, and the administrative assistance from Lynne Hollander, Betty Riccio, Monica Wong, Irene Zagazeta, Barbara Morgan, and Kevin Lauderdale. Larry Fagan, Tom Rindfleisch and Rosalind Ravasio provided a perfect environment for the development of research, as well as personal encouragement for my project. Samson Tu and Bill Yeager were always ready to help. Lyn Dupré taught me how to write clearly, and Debby Fife made sure I did it. Holly Jimison, Jaap Suerdmont, and John Egar made me feel at home as soon as I arrived at Stanford, and have always given me great advice. My fellow colleagues and friends Gretchen Purcell, Bill Detmer, Rhea Trombropolous, Alex Poon, Sue Henry, Manon Kuilboer, and Chris Davidson helped the transition “back to school” to be smooth and fun, and all students in the program helped the years of research to be nice, stimulating, and, as much as possible, enjoyable.

I want to acknowledge the special help I received from Pat Swift and Darlene Vian, who really made me see beyond this dissertation and be able to have a glimpse of the “big picture.” It is people like them who make this place so special. There is no possible way I could thank them enough for their friendship and precious advice, except to mention that I need no other role models.

I would not have gotten here without everything I learned from my teachers and friends in Brazil, nor without the continuous support of my mother and brothers, for whom the burden of geographical distance was compensated by the feeling that I was doing

something I really enjoyed. The love, understanding, and sense of humour of my husband, Ruy, made this work possible and worthwhile.

There is a tremendous sense of achievement in completing this work. It is one dream that came true and will always remind me of the happy years we spent in Palo Alto. We are now leaving this place with much more than a few suitcases and dreams. I have learned many things, and I am carrying in my arms a most special dream, literally born at Stanford: my son, Thomas, to whom I dedicate this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 Introduction

The field of medical informatics has evolved around structuring, processing, storing, and transmitting medical information for a variety of purposes [Shortliffe, 1990]. One of these purposes is to develop decision-support systems that enhance the human ability to diagnose, treat, and assess prognoses of pathologic conditions. Even if disease processes were fully understood, population variability would still make individualized diagnosis, treatment, and prognosis—all essential parts of good health care—difficult classification tasks. The reality is, however, that diseases are not fully understood, nor is population variability fully taken into account in many decision-making situations. Sometimes it is not possible for a clinician to employ the principles learned in the basic and clinical sciences to determine whether a patient has a given disease, whether he or she should be given a certain treatment, and how long he or she will survive.

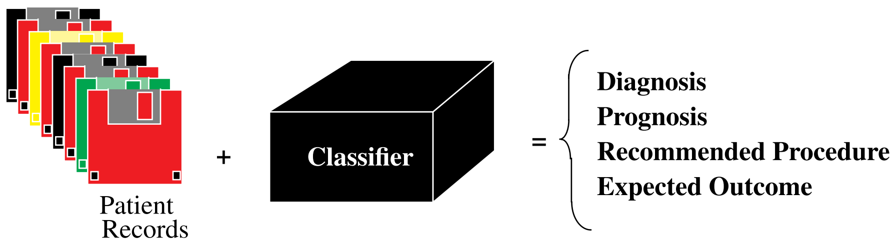
Studies that use aggregate data provide a "statistical rational" that often overcomes the limitations of reasoning from first principles. These studies are not only abundant in the medical literature, but are also key in defining practice guidelines for diagnosis and treatment and in defining prognostic indices. The generality of these studies may inhibit their being used in a practical setting, where senior clinicians still emphasize that a case-by-case analysis is always necessary, and therefore employ a routine that resembles the classic nearest-neighbor algorithm [Duda, 1973]: Use the results of the study whose population characteristics most closely resemble the patient at hand. The ideal situation when decision-support tools are employed, however, would be to get the most likely diagnosis, prognosis, or ideal treatment *for the particular case at hand*.

Current statistical models based on regression offer the possibility of establishing individualized responses, but may require unrealistic assumptions about the distribution and interdependence of data or errors. Oversimplified models constructed according to such assumptions may be of limited value. Furthermore, algorithms that are able analytically to calculate exact solutions for certain simplified problems are gradually being replaced by algorithms that utilize numerical methods to reach approximate solutions to real-world complex problems [Maron, 1987]. Neural networks (see Chapter 2) have been shown to solve complex problems with high accuracy. They constitute good alternatives to current regression models in medicine, although they have some drawbacks. One of these drawbacks is that certain types of neural networks are very slow in recognizing infrequent patterns, and often cannot recognize these patterns at all. Rare patterns do occur, however, and must be learned if practical application of neural networking technology is to be achieved.

1.1 Problem: Recognition of Infrequent Patterns

Researchers in medical informatics have dealt with the problem of encoding clinical data for electronic processing for a long time. These data can be used for a variety of purposes, including automated recognition of patterns and classification by machine-learning methods (useful in making diagnoses, predicting prognoses, recommending procedures and treatments, and forecasting outcomes), as depicted in Figure 1.1.

Figure 1.1. Machine learning in medicine.



Electronic medical data can serve as input to classifiers such as neural networks. In these machine-learning approaches, models are constructed from available data by the computer. Outputs are classifications.

1.1.1 Machine learning and medicine

Machine-learning methods for classification provide inexpensive means to perform diagnosis, prognosis, or detection of certain outcomes in health care research. With the increasing number of electronic clinical databases, and the increasing costs of manual processing, it is likely that machine-learning applications will be necessary to detect rare conditions, unnecessary procedures, and unexpected outcomes. Although these patterns are infrequent, they may be typical and their detection is important. Therefore, there is a need to enhance the predictive power of machine-learning methods by increasing sensitivity for low-frequency patterns without decreasing specificity.

For example, let us suppose that a South American patient comes to a California clinic with signs, symptoms, and test results that point to cardiomegaly, megaesophagus, and megacolon. The physician may recognize the pattern of Chagas' disease[1] immediately, even though this disease is extremely rare in the U.S. If a large data set of patients coming to the same clinic were available, machine-learning methods could be used to develop systems that assist with the diagnoses of diseases for patients presenting in that setting. The medical researcher would want Chagas' disease to be recognized if it were sufficiently different from other diseases, even though its prevalence in the data set would be very low.

[1] Chagas' disease is caused by *Trypanosoma cruzi*, and is common in southern Brazil and northern Argentina. Some cases have been described in the southern United States and Mexico [Veronesi, 1992].

My goal in this research is to provide a means for improving the ability of machine-learning methods to recognize infrequent patterns and thereby improve their predictive abilities. I focus on learning using neural network technology and, in particular, the method known as *backpropagation*.

A large number of medical applications in which classification is desired have the goal of discriminating a pattern with low frequency (e.g., “thyroid disease,” “bad prognosis”) from a pattern with high frequency (e.g., “no disease,” “good prognosis”). For example, if only a very small group of patients who have undergone bypass surgery have prolonged lengths of stay in hospital, this category will hardly be recognized by most machine-learning methods. These patients, however, provide exactly the patterns that need to be studied and followed more closely. Another example is screening for certain diseases with low prevalences but for which there is some form of intervention that will improve the patient’s or the population’s well-being, and the overall benefit of detecting a case justifies the costs (e.g., screening for congenital hypothyroidism, a disease that has a prevalence of 1/4,000) [U.S. Preventive Task Force, 1989].

Even though the patterns for certain conditions and diseases are well known to the medical community, others may be not as well defined. For example, we still do not know why some patients with the HIV virus survive longer than others who have been infected for similar periods of time. Screening electronic databases of HIV-infected individuals may provide some clues regarding how these patients cluster, and establish a means to predict prognosis for such clusters. Furthermore, certain cases represent patterns that will characterize a condition only if sufficient numbers of such cases are reviewed. Rare conditions or unexpected outcomes may constitute only one or two cases in the career of a health care provider. Instead of discarding such cases as mere “outliers,” pooling them in an electronic database and using a machine-learning method may reveal interesting and important correlations. As structured electronic medical records become more common and more widely accessible to researchers, screening large data sets for certain patterns (also called “database mining for knowledge discovery” by computer scientists) may be greatly enhanced by the use of machine-learning methods. The patterns detected by these methods can then be processed by a number of manual or computer-based decision-support applications.

1.1.2 Survival analysis

Survival analysis can be considered a classification problem in which the application of machine-learning methods is appropriate. In this case, the outputs of a classification system are categories that correspond to predetermined intervals of time. A prognostic estimate may be produced for each interval of time. Although the results of such a classifier may not seem as precise as those of classical models of survival analysis, the final use of those results is probably the same. For example, if a patient is told that his mean survival time is 198 days, plus or minus 12, he will translate this information to "My mean survival is approximately six months." By establishing meaningful intervals of time according to a particular situation, survival analysis can easily be seen as a classification problem.

Survival analysis plays an important role not only for healthcare policymakers, but also for the clinician. The results of survival analyses can be used for individual prognosis (as in the case of an AIDS patient who wants to know how long he is likely to survive), for population prognosis (as in the case of a health minister in Africa, who wants to know how many adults will compose the work force in the next decade), or even for commercial interests (as in the case of a pharmaceutical company that wants to know how much zidovudine to produce next year).

In chronic diseases, such as myocardiopathies, the number of individuals who die within a certain period of time, compared to the pool of all diseased subjects, is small. Predicting death for such individuals may be important both at the individual level of patient care and for health policy planning. Survival analysis applies not only to the study of a deadly event, but also to the study of the duration of a normal condition, such as "healthy status." Therefore, the development of a certain pathologic condition after a period of time (e.g., myocardial infarction) is also amenable to survival analysis. Even in a disease with a

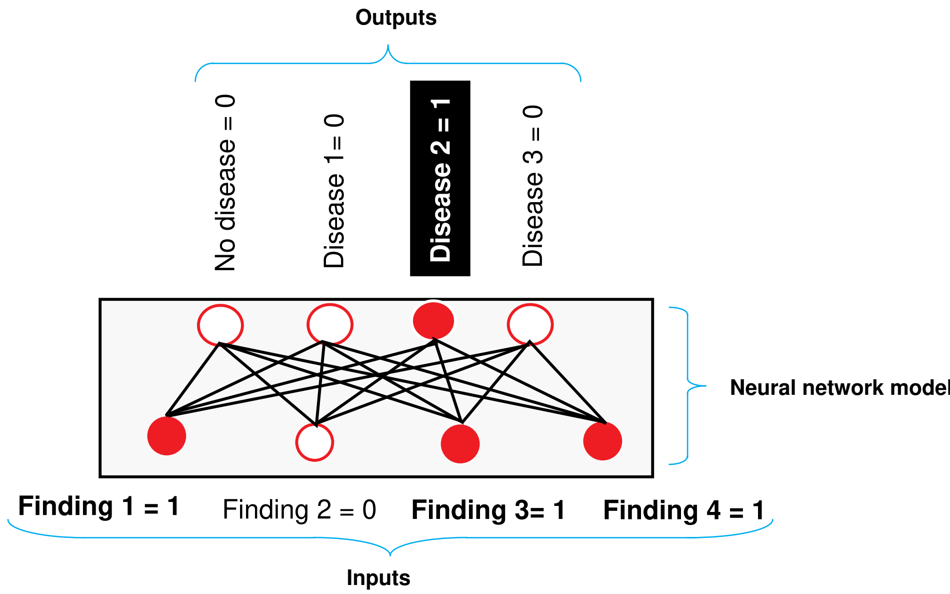
100 percent fatality rate, such as AIDS, the number of individuals who die in the first year after infection is low, compared to the set of all individuals with the disease. It is therefore necessary to be able to recognize low-frequency patterns in survival analysis. Classical statistical methods of survival analysis require certain assumptions about the distribution of the data (e.g., normal distribution, proportional hazards, as discussed later in Section 5.2). Neural networks can constitute a good alternative when some of these assumptions cannot be verified.

1.1.3 Neural networks

Neural networks, also known as connectionist or parallel distributed processing (PDP) systems, are machine-learning models implemented using a computational framework developed primarily to understand and simulate physiological neural systems [Rumelhart, 1986]. While neuroscientists still utilize neural networks to simulate the function of real neurons, engineers, analysts, physicians, and scientists have used them to model processes as diverse as the classification of military targets, the prediction of stock market activity, the recognition of speech, and the diagnosis of medical problems [Hertz, 1991].

Neural networks may have the same inputs and outputs of a regression model, and may be built to perform exactly the same tasks. For example, Figure 1.2 shows a hypothetical neural network set up to diagnose four conditions from cases that have information on four findings.

Figure 1.2. Hypothetical neural network for diagnosis.



Inputs for this simple neural network include the presence or absence of four findings. In this example, Findings 1, 3, and 4 are present, and Finding 2 is absent. The network provides the diagnosis of Disease 2. The inputs and outputs of this neural network model could be used in a regression model for diagnosis as well.

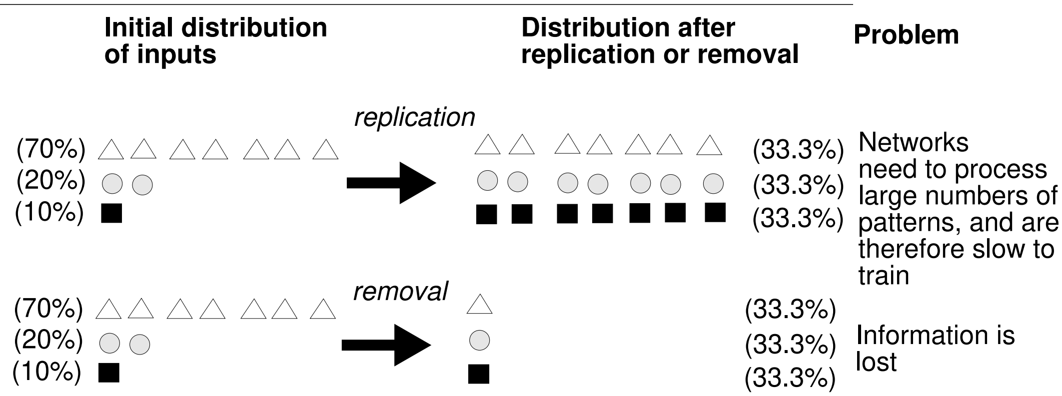
Neural networks have been applied for a variety of purposes in biomedical research. A search of the MEDLINE medical literature database for the years 1982 to 1995 yields more than 600 articles that describe neural networks for diagnosis, prognosis, or clustering of medical data, and applications in the basic medical sciences, especially molecular biology. In Chapter 2, I provide a summarized history of the development of neural networks, and their application in many biomedical domains. I explain how neural network systems work and how they acquire knowledge from training examples.

The *backpropagation algorithm* for estimating parameters in neural networks[2] has been the most popular in the medical literature [Reggia, 1993], and it is explained in Chapter 2. One of the problems encountered by researchers utilizing the backpropagation algorithm is that low-frequency patterns (or rare patterns) may take a long training time to

[2] Estimating parameters in neural network models is also called *learning* or *training* in the neural network literature.

be recognized, because frequent patterns “dominate” the error. Some rare patterns cannot be recognized at all. In some biomedical applications, the pattern of interest *is exactly the one that is rare*, and backpropagation-based neural networks may have difficulties in learning this pattern. In survival analysis, the final event (usually death) is relatively infrequent in each time interval, so the use of neural networks for survival analysis has been limited. The difficulties related to learning infrequent patterns in neural networks have led some investigators to propose algorithms for preprocessing the data and to develop modifications of the backpropagation algorithm. One of these solutions has been the *replication* of rare patterns in the training set (or the *removal* of some instances of the most frequent pattern), such that all categories become equally represented.[3]

Figure 1.3. Equal representation of patterns to neural networks.



Replicating infrequent patterns so that all output categories become equally represented in the training set leads to an increased number of training cases, which makes training slower. Removing frequent patterns causes the loss of information contained in removed cases.

Figure 1.3 shows the methods for assuring that the neural networks receive an equal number of instances of several patterns. In the case of *replication*, the number of training patterns may become so large that training gets extremely slow. Furthermore, information on prior probabilities is ignored. In the case of *removal* of frequent pattern instances, important information contained in the removed cases is lost.

Another solution to the problem of dealing with infrequent patterns in

[3] D.E. Rumelhart. Personal communication, 1994.

backpropagation-based neural networks has been the modification of the weight update function used in the backpropagation algorithm, such that utilities for identifying specific patterns are taken into account in the training phase [Lowe, 1990]. In this *utility-modified backpropagation* method, some patterns are deemed more important than others, so the utility of recognizing them is embedded in the backpropagation cost function.

These existing solutions imply either a significant preprocessing or manipulation of the data (*replication* or *removal*), which leads in many cases to changes in the prior probabilities of each output category in the transformed data sets, or to a significant change in the backpropagation algorithm (*utility-modified backpropagation*), with a degree of custom-tailoring that makes it necessary to retrain the network every time utilities change. Both solutions significantly increase the neural network model's sensitivity for rare patterns at the expense of a concomitant significant decrease in its corresponding specificity or an increase in learning time. Chapter 3 is dedicated to the problem of recognizing rare patterns in neural networks whose training is based on the backpropagation algorithm, and to the comparison of existing solutions. In Chapter 4, I present another solution: the use of hierarchical systems of neural networks. This solution can be generalized to a sequential system of neural networks that has special use in survival analysis, as shown in Chapter 5.

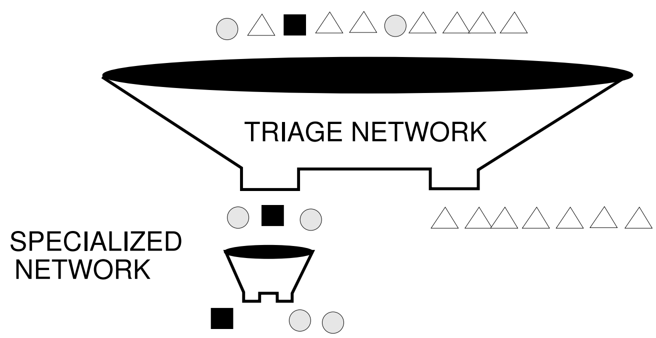
1.2 Solution: Hierarchical and Sequential Systems of Neural Networks

A relevant issue for the use of predictive models in health-related research, which is not addressed fully by the classical statistical classifiers, is the ability to perform hierarchical classification. In medical practice, this type of prediction or classification, as opposed to a one-step procedure, is often desirable given the time constraints and the nature of the medical interventions. For example, the health care worker may need to take actions before a certain time (e.g., the patient's next appointment), or want to know only a patient's short-term prognosis for developing a given infection, so that adequate prophylactic drugs can be prescribed. It may even be unnecessary for the healthcare worker to reach a definitive or final diagnosis if the treatment for all entities in a given differential diagnosis is the same. Analogously, it may be unnecessary to know the long-term prognosis if there are no ways to influence the long-term outcome. By narrowing the differential diagnosis at an early stage, the health care worker can avoid ordering unnecessary, expensive, and potentially harmful tests. For survival analysis, a precise prediction of the short-term prognosis for a given patient may help the health care worker to select the best therapeutic measures and to discard expensive and invasive interventions that have little chance of being successful. A full battery of tests that help to predict long-term prognosis or assist in the choice of interventions that have only long-term effects may not be needed, for example, in a case where the patient has a poor short-term prognosis. Instead, resources may be allocated to try to reverse the causes for the unfavorable prognosis at the early stage, or at least to enhance the quality of life for that patient.

Hierarchical classifiers are not common, but they can be implemented within most machine-learning models. They should not be confused with other classification methods that partition the outcome space according to a small number of variables, as do recursive partitioning methods [Breiman, 1984]. Hierarchical classifiers can partition the outcome space according to multiple variables. There are several advantages to having an automatic classifier perform hierarchically. First, development and implementation of the model can be incremental; that is, detailed classification can be postponed to a later stage. Second, there is a potential for identifying where in the hierarchy the classifier starts to lose its discriminating power. For example, a classifier that determines whether a patient has a disease that belongs to a large class of diseases, such as *hypothyroidism,* is more accurate than a classifier that determines whether the patient has a specific type of hypothyroidism. In this case, the hierarchical system may start to lose its discriminating power after the classification of *hypothyroidism* is achieved. Third, the use of the full set of attributes may be unnecessary at different levels, so censored data may be used. For example, data from patients who do not have measurements from a certain laboratory test should not be discarded in the initial phase if the missing test is necessary only to define a detailed and final classification.

The hierarchical architecture of neural networks that I propose is depicted schematically in Figure 1.4, and is described in detail in Chapter 4. In hierarchical systems of neural networks, a triage network divides the sample into smaller groups, classifying the cases according to similarity and creating abstract groupings. The smaller groups constitute inputs to specialized networks that are able to discriminate certain patterns with enhanced accuracy and at enhanced speed, as will be demonstrated in Chapter 4.

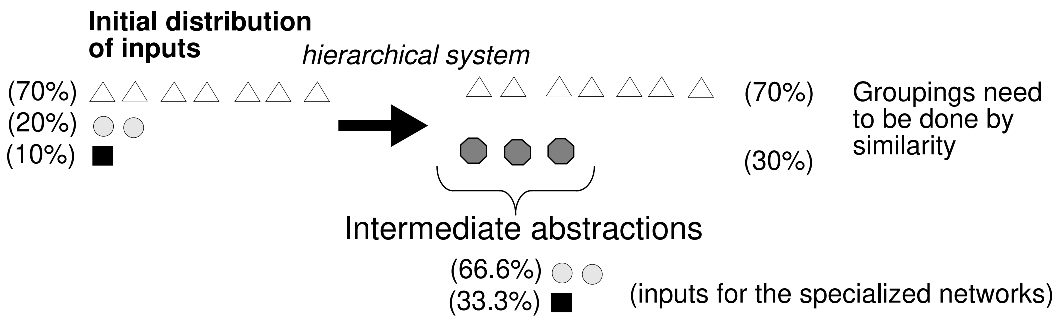
Figure 1.4. Hierarchical system of neural networks.



A triage network is used to "filter" interesting cases (represented by squares and circles) from the whole training set. The filtered instances are further processed by specialized networks that provide the final classification.

Hierarchical neural networks (HNNs) do not imply a change in the backpropagation algorithm per se, but they provide a method for constructing and training neural networks incrementally. The backpropagation algorithm is utilized in its pure form in each of the various levels of the hierarchical system. There is no need to alter the weight update function for each output category. This method separates the process of categorizing using input features from that of assigning utilities for each correct classification to obtain the best decision boundary based on a decision-theoretic principle. In addition, the preprocessing of data to form abstractions used in the intermediate levels of the hierarchical system may provide a means to later explain the system's reasoning. Patterns are grouped by similarity, and the changes in prior probabilities are due solely to rearrangement of patterns in similar groups. There are no replications or deletions of data that change the prior probability of each output category in the system as a whole. Figure 1.5 shows how this grouping permits an increase in frequency of the patterns that will be processed in specialized neural networks.

Figure 1.5. Representation of patterns in the hierarchical system.

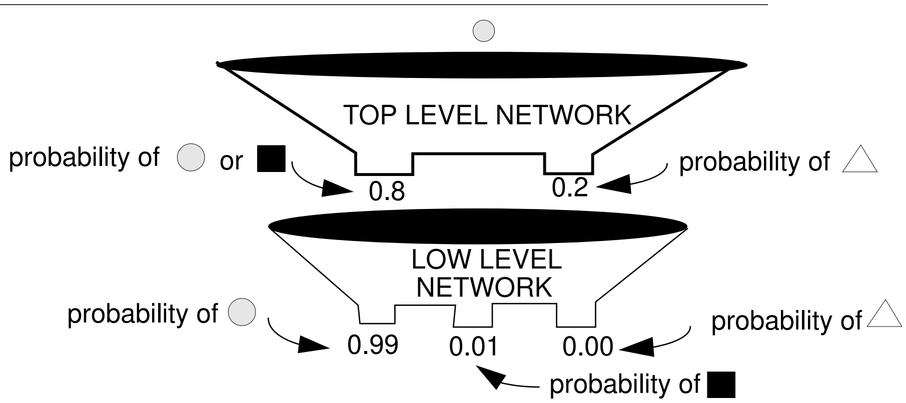


In the hierarchical system, patterns are grouped in intermediate abstract groups, based on similarity. The patterns that constitute the intermediate abstractions are then processed by their own specialized neural network. By removing frequent patterns from the whole training set, patterns that were relatively infrequent in the initial set become more frequent as inputs to the specialized networks.

Related research on hierarchies of neural networks is discussed in Chapter 4.

Sequential neural networks, shown in Figure 1.6, are similar to HNNs.

Figure 1.6. Sequential system of neural networks.

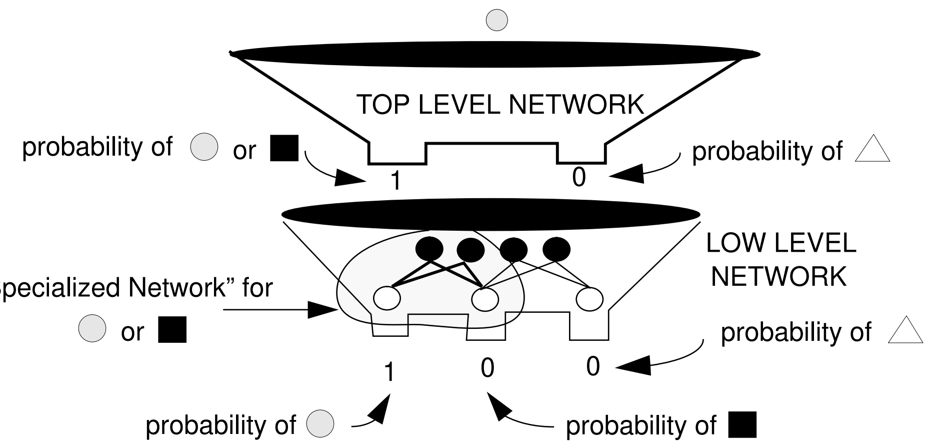


A top-level network is used to provide a probability that a case belongs to a certain class. The low-level network uses this information to provide the final classification.

The main difference is that, instead of triaging cases into two or more classes, the top-level network provides a probability that the input belongs to any of the classes (intermediate abstractions). The low-level network processes all cases, but it is given a "hint" about which class a given case belongs to. This decomposition facilitates the discrimination of infrequent patterns by making the classifier focus its attention on a certain range of prognostic indices, as explained in Chapter 5.

HNNs can be viewed as special cases of sequential neural networks, in which the top-level network makes a binary decision on whether the case belongs to a certain class, sending either a “0” or a “1” to the low-level network. Depending on the classification at the top level, the low-level network behaves differently, utilizing the portion of the low-level network that specializes in cases of that class, as shown schematically in Figure 1.7.

Figure 1.7. HNNs are special cases of sequential neural networks.



1.3 Hypotheses

The goal of this research is to show that neural networks can make an accurate individualized prognosis of a patient given his or her particular condition. For example, several studies show that the survival of patients diagnosed with AIDS for more than five years is an unlikely event. However, this statistic applies to the whole pool of patients. If we have more information about a particular patient (other than just the date of the diagnosis), such as age, gender, AIDS-defining diagnosis, and laboratory test results, we may be able to predict the prognosis for that patient more accurately. Special systems of hierarchically or sequentially arranged neural networks, described in detail in Chapters 4 and 5, were built for the experiments described in this dissertation.

I tested the hypotheses that (1) sequential systems of neural networks produce results that are more accurate than are those of nonsequential neural networks (in terms of calibration and resolution, discussed in Chapter 6), and (2) in certain circumstances, neural networks produce more accurate estimates of survival time than Cox proportional hazards models and logistic regression models. The hypotheses were tested in the real data sets described in Chapters 7 and 8. As the results show, sequential neural networks outperformed nonsequential neural networks and Cox proportional hazards models.

1.4 Validation in Medical Data Sets

The use of clinical or epidemiological data sets for developing and testing neural network models imposes difficult challenges. Typically, these data sets do not contain thousands of cases, nor do they constitute complete and noise-free collections. Furthermore, some categories of data may be underrepresented, making the learning process slow. There are currently no published guidelines for deciding when to use a neural network, nor are there guidelines for which network architecture to use in a given setting. HNNs have been successfully applied in studies using an artificial data set and a large clinical data set of patients suspected of having thyroid diseases. These studies are presented in Chapter 4.

The sequential neural network model, presented in Chapter 5, was evaluated in two different sets of medical data: (1) the AIDS Time-Oriented Health Outcome Study (ATHOS) data set (a data set of HIV+ patients whose death over each interval of a long follow-up period is the low-frequency event) [Fries, 1992] and (2) the Framingham data set (a data set of patients regularly followed for several years by researchers interested in prospectively investigating the epidemiology of cardiac diseases) [Dawber, 1980]. The low-frequency episode in the latter case is development of coronary heart disease (CHD). These data sets encompass different challenges: the ATHOS data set is a small collection of cases that contains many missing values and addresses a medical problem that is relatively new; the Framingham data set is a collection of several cohorts of patients who may or may not develop a highly prevalent condition (cardiac disease) the contributing factors of which are still under study. The results of this work may serve as a benchmark for the performance of the different prognostic models in these data sets. Using the ATHOS data set, I tested the hypothesis that sequential neural networks can be used to predict death due to AIDS, and that its results are better than those of nonsequential neural networks and the Cox model. Using the Framingham data set, I tested the hypothesis that sequential neural networks can adequately model CHD development, and that their performance is better than that of logistic regression models. I also compared standard and sequential models.

Chapter 7 describes the Framingham data set and discusses the relevance of novel neural network models of the development of cardiac conditions. Chapter 8 describes in detail the ATHOS data set and the significance of improving prognostic methods of HIV progression and survival modeling for patients with AIDS. In each of these chapters, specific hypotheses are stated, and specific evaluation procedures are explained. The gold standard for the evaluation was the actual data contained in the data sets. The performance was evaluated by comparing calibration and resolution.

1.5 A Guide to the Reader

In Chapter 2, I review the history of neural network development and describe selected applications of neural networks in biomedicine.

In Chapter 3, I discuss the problem of learning rare patterns in medical data, and I compare current approaches for dealing with this problem. I use an artificial data set to illustrate the problem.

In Chapter 4, I present an architecture of hierarchical neural networks that can be used to address the problem described in the previous chapter, and I compare its performance with the current models using the data set presented in Chapter 3. I present also an example of a diagnostic task that is facilitated by the use of HNNs in the domain of thyroid diseases.

In Chapter 5, I discuss additional functionality required by prognostic systems. In these types of problems, noise is present, missing values are abundant, and the boundaries of each output category are sometimes not well delimited. I present an architecture of sequential neural networks (of which HNNs are a special case) and its use in survival forecasting.

In Chapter 6, I review the evaluation methods required for assessment of performance of prognostic systems.

In Chapter 7, I describe the neural network and the logistic regression models that were used to predict CHD in the Framingham data set. Dependent and independent variables, as well as model specifications, are presented in detail.

In Chapter 8, I describe Cox proportional hazards and neural network models used to predict death due to AIDS in the ATHOS data set.

In Chapter 9, I discuss the implications of the results in both data sets, highlighting the differences and similarities of both experiments, and generalizing the conclusions.

In Chapter 10, I provide a summary of the dissertation, its contributions to the field of medical informatics, and the lessons learned in this research. I also comment on future extensions of this work.

CHAPTER 2

Neural Network Applications in Medicine

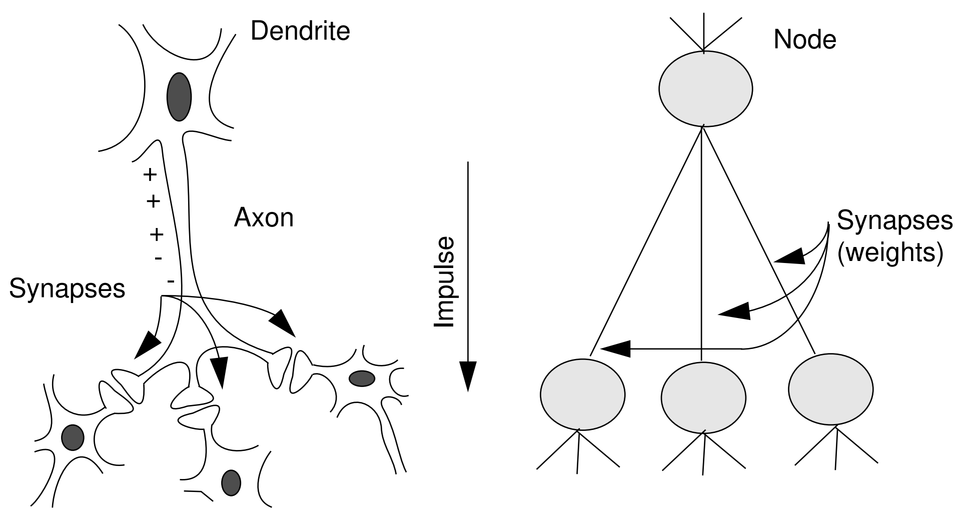
The purpose of this chapter is to provide an overview of neural networks and their applications in several areas of medicine. The most frequently used algorithm for neural network learning in medical applications, the backpropagation algorithm, is presented in detail. A summary of the development of neural networks and the concepts of supervised and unsupervised learning is presented in Section 2.1. Section 2.2 summarizes some applications of neural networks in medicine, explaining how neural networks can be used as statistical tools for making inferences and in which aspects they are more promising than conventional statistical techniques. In Section 2.3, guidelines for evaluating neural networks in medicine are suggested.

2.1 Brief Introduction to Neural Networks

2.1.1 History

Neural networks, also known as connectionist systems or parallel distributed processing models, are computer-based, self-adaptive models that were first developed in the 1960s, but they reached great popularity only in the mid-1980s after the development of the backpropagation algorithm by Rumelhart et al. [1986]. Initially derived from neuroscientists' models of human neurons, neural networks now encompass a wide variety of systems (many of which are in no way intended to mimic the functions of the human brain). Neural network research has its origins in the work developed by McCullough and Pitts [1943], who developed mathematical models based on observational studies of real neurons. Figure 2.1 compares the anatomies of real and artificial neural configurations.

Figure 2.1. Real and artificial neural networks.

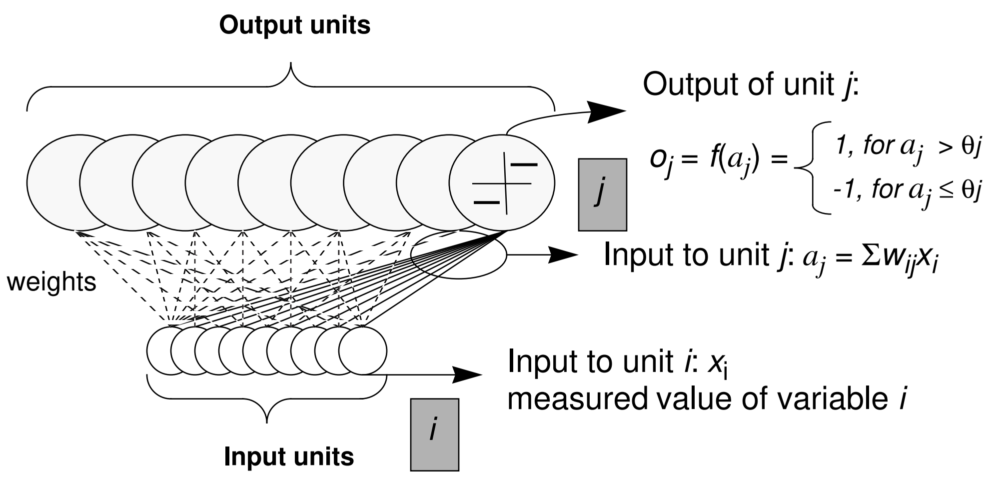


The neural body is represented in artificial neural networks as a circle, and is called a *node*. The synapses are represented as lines connecting nodes, and are called *weights*.

In artificial neural networks, the connections are called *weights* and are represented by real numbers. The Hebbian rule [Hebb, 1949] for learning in simple neural models dictates that, if two connected neurons (or *nodes*, in the case of artificial neural networks) are simultaneously in an active state, the connection between them should be strengthened. Making a connection between two nodes stronger means that the real number is increased by a certain positive amount. Making a connection weaker requires that a negative number be added to the weight. Common learning rules used in artificial neural networks are derived from the Hebbian rule.

A perceptron is the simplest form of a neural network model. It is composed of an input layer (where values for attributes are entered) and an output layer (where output values are produced). Figure 2.2 shows a simplified version of a perceptron. Input values for attribute i of a certain pattern are represented by x_i, weights connecting units i and j are represented by w_{ij}, net inputs to unit j are represented by a_j, thresholds (also called *biases* in the neural network literature) for unit j are represented by θ_j, and the output from unit j is designated o_j (which is a function of a_j and θ_j).

Figure 2.2. Fundamental elements of a perceptron.

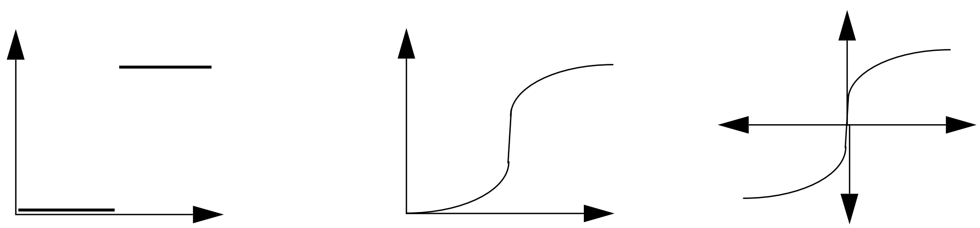


Perceptrons are composed of an input layer and an output layer. Input values are multiplied by weights and the result constitutes the input for the output layer. In the output layer, an activation function will determine the activation of the output node for a given input pattern.

2.1.2 How neural networks work

Each layer in a neural network is composed of several nodes, each of which has an associated activation status that is a function of the node's input value. Each node or unit in the output layer receives inputs from the incoming connections, processes the value with an activation function (also called transfer, squashing, or gain function), and produces its own output, which represents the activation status. Commonly used activation functions are the logistic, the linear threshold (usually used in perceptrons), and the hyperbolic tangent, shown in Figure 2.3.

Figure 2.3. Activation functions.



Linear Threshold

$$Activation = \begin{cases} 0, \text{ if } a \le \theta \\ 1, \text{ if } a > \theta \end{cases}$$

Logistic

$$Activation = \frac{1}{1 + e^{-a}}$$

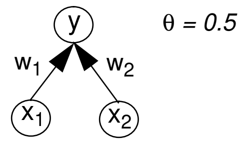
Hyperbolic Tangent

$$Activation = \frac{e^{a} - e^{-a}}{e^{a} + e^{-a}}$$

Initial neural network models used the linear threshold function to simulate the behavior of real neurons, which are only active if the impulse they receive is above a certain threshold. Logistic and hyperbolic tangent functions were used to make the function differentiable, and differ only in the output range (0 to 1 for the logistic and -1 to 1 for the hyperbolic tangent).

Figure 2.4 shows an example of a neural network that performs the Boolean function AND. The inputs and outputs are composed of binary digits. Whenever "00," "01," or "10" are presented, the output is "0," and whenever the input "11" is presented, the output is "1." The neural network has to set up values for its weights that will always reproduce these results. The threshold (or bias) for the output node determines the value over which the node will start to produce an output of "1" instead of its default "0." That is, whenever the output node receives an input over 0.5, it will produce a "1."

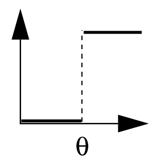
Figure 2.4. Simple neural network that performs the Boolean function AND.



input	output
00	0
01	0
10	0
11	1

$f(x_1w_1 + x_2w_2) = y$

$f(0w_1 + 0w_2) = 0$
$f(0w_1 + 1w_2) = 0$
$f(1w_1 + 0w_2) = 0$
$f(1w_1 + 1w_2) = 1$



$$f(a) = \begin{cases} 1, \text{ for } a > \theta \\ 0, \text{ for } a \leq \theta \end{cases}$$

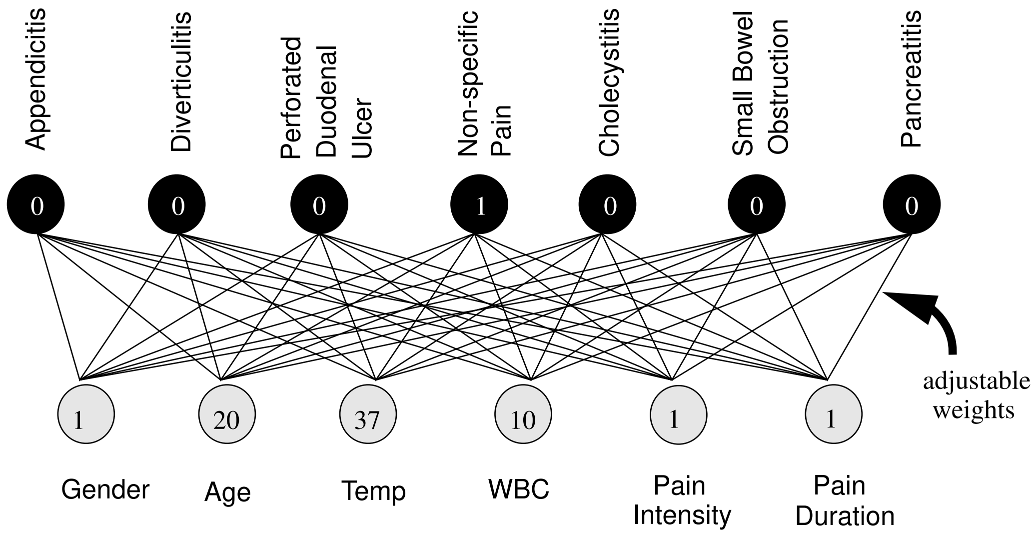
some possible values for w_1 and w_2

w_1	w_2
0.10	0.45
0.20	0.40
0.25	0.30
0.40	0.20

Input units receive values from x_1 and x_2, which are multiplied by w_1 and w_2. The result will be the input to the output unit, and will be entered in a threshold function that has a bias $\theta = 0.5$. If the input is greater than 0.5, the output unit will produce a “1”; otherwise, it will produce a “0.” Some possible values for the weights w_1 and w_2 are displayed in the lower table. The neural network learning algorithm determines best values for the weights.

A neural network may have many output units. Usually, the output unit that has the highest activation at the end of the training phase will indicate the predicted category. In a classification application, inputs are generally composed of the attributes of each instance in a data set, and outputs constitute classification categories. For example, a perceptron that was designed to classify patients complaining of abdominal pain is shown in Figure 2.5. In this example, each case is represented by a set of attribute values, which may be continuous, such as “temperature,” or discrete, such as “male.” In this example, the perceptron concludes that the patient has “non-specific abdominal pain.”

Figure 2.5. Perceptron for diagnosing abdominal pain.



In this hypothetical example, the goal is to diagnose conditions related to abdominal pain. Inputs correspond to demographic data or measured values for *Age, Temperature*, etc. The initial random weights will be adjusted in order to produce correct diagnoses for a given training set.

Neural networks can classify patterns quickly once they know the values of the weights, by performing simple operations such as multiplication and addition. However, learning the values of the weights may take a long time, as will be shown next.

2.1.3 How neural networks learn

Learning in neural networks is performed by iteratively modifying weights such that the desired output is eventually produced by the network, with a minimal amount of error. Typically, initial small random weights are updated gradually.

Going back to the example in Figure 2.4, suppose that the network started with random weights of $w_1 = 0.9$ and $w_2 = 0.85$. The initial outputs for patterns “00,” “01,” “10,” and “11,” using the threshold or bias θ of 0.5 and a linear threshold activation function *f,* would be

$$f(x_1 * w_1 + x_2 * w_2) = y$$

$f(0*0.9 + 0*0.85) = 0$

$f(0*0.9 + 1*0.85) = 1$

$f(1*0.9 + 0*0.85) = 1$

$f(1*0.9 + 1*0.85) = 1$

respectively. In this case, it is easy to see that the network should decrease its weights in order to produce the desired response. The sum of w_1 and w_2 must be greater than 0.5, and both w_1 and w_2 should be smaller than 0.5. Taking the difference between the desired output and what the network produced for each pattern gives us an error of -2 (since patterns “01” and “10” are wrongly producing high results at this point). The next step is then to change w_1 and w_2 in the direction that minimizes the error. A small change is done at each cycle of the network training phase, which is guided by the direction (signal) of the error, and a constant of proportionality η (or *learning rate*), according to

$$\Delta w = \eta \delta a$$

where δ is the difference between target (desired output) and real output, and a is the value entered in the input unit. The updated weight is calculated by summing the Δw and the initial weight. Therefore, if η is 0.3, the weights w_1 and w_2 change to

$0.90 + [0.3 * (0 - 0.9) * 1] = 0.63$

and

$0.85 + ([.3 * (0 - 0.85) * 1] = 0.595$

respectively. These updated weights will produce

$f(0*0.63 + 0*0.595) = 0$

$f(0*0.63 + 1*0.595) = 1$

$f(1*0.63 + 0*0.595) = 1$

$f(1*0.63 + 1*0.595) = 1$

The solution was not achieved at this first training cycle (or *epoch*), so another cycle begins, where weights w_1 and w_2 are changed to 0.63 - 0.189 = 0.441 and 0.595 - 0.1785 = 0.4165. This set of weights solves the problem, since

$f(0*0.441 + 0*0.416) = 0$

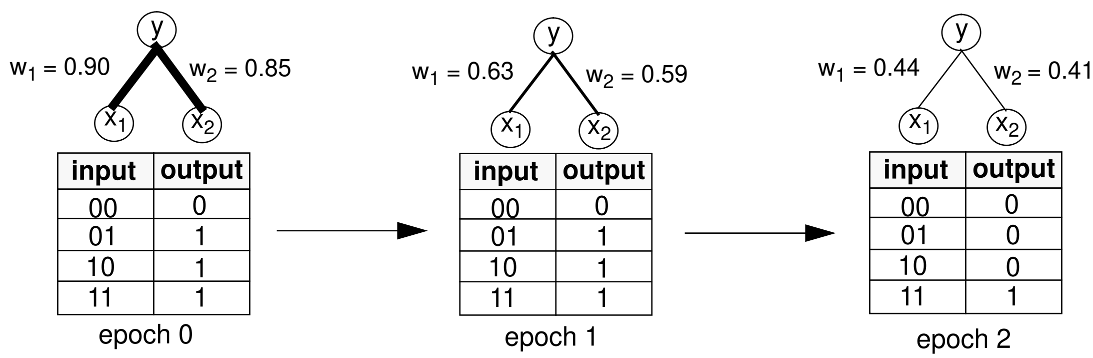
$f(0*0.441 + 1*0.416) = 0$

$f(1*0.441 + 0*0.416) = 0$

$f(1*0.441 + 1*0.416) = 1$

No more training cycles are necessary. Figure 2.6 shows the weight changes.

Figure 2.6. Weight changes for the example in Figure 2.4.



In this example, it was easy to see that the weights should be decreased, since the outputs were too high for patterns “01” and “10.” In practice, however, knowing how to change the weights is much less obvious.

Widrow and Hoff [1960] developed what is nowadays called the *delta rule* for estimating parameters of their models of one-layered neural networks (called *Adalines*, or adaptive linear elements). Adalines are very similar to perceptrons, but they use the logistic activation function in the output layer. The delta rule performs the parameter (weight) estimation iteratively. The rule is simple: Whenever the network’s output is not close enough to the desired output, a change in weights occurs in the direction that minimizes the error. The change is proportional to the difference between the network’s output and the desired output, or target.

The cost function being minimized is usually

$$E \propto \sum_p \sum_i (t_i - o_i)^2$$

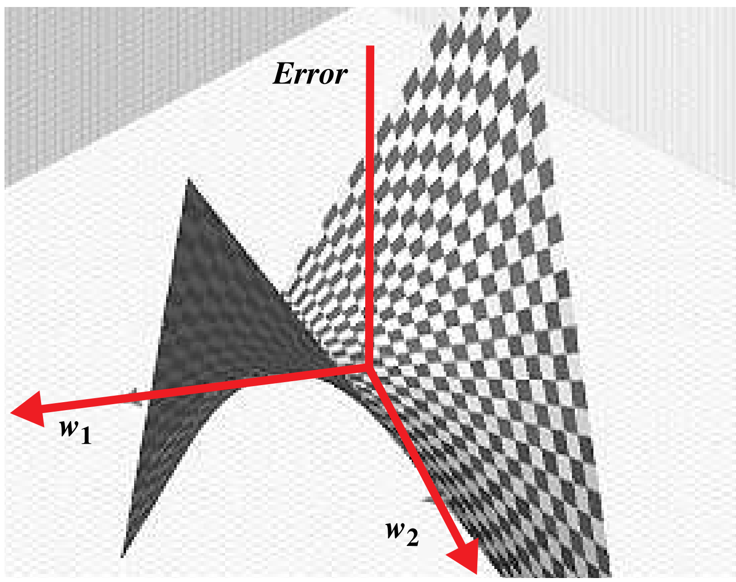
or

$$E_p = \frac{1}{2} \sum_i (t_i - o_i)^2$$

where E is the cumulative error, t is the target, or desired output, for each pattern p and each output unit *i,* and *o* is the real output [Hertz, 1991]. In other words, the error is a function of the difference (or *delta*) between what the network produced and what we wanted it to produce. Other error functions, such as the cross-entropy error, have also been used [Curry, 1990].

Learning in neural networks means finding a set of weights that minimizes the overall error. Figure 2.7 shows an error surface defined by two weights. The objective of learning is to find the lowest location in the error surface, by modifying the weights in the direction that minimizes the error.

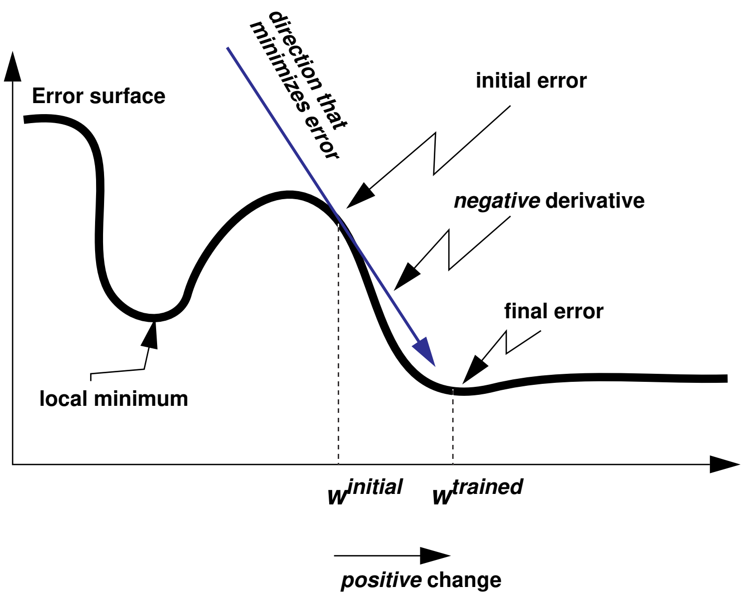
Figure 2.7. Error surface.



Learning is done by modifying the weights w_1 and w_2, so that the *Error* decreases.

A common way to find a suitable set of weights that minimize the error (at least locally) is to perform *gradient descent*, i.e., to modify the weights such that the changes are inversely proportional to the derivative of the error with respect to the weights [Rumelhart, 1986]. For example, Figure 2.8 shows the direction of weight change in a problem where there is only one input, and therefore only one weight.

Figure 2.8. Error change according to the derivative of the error.

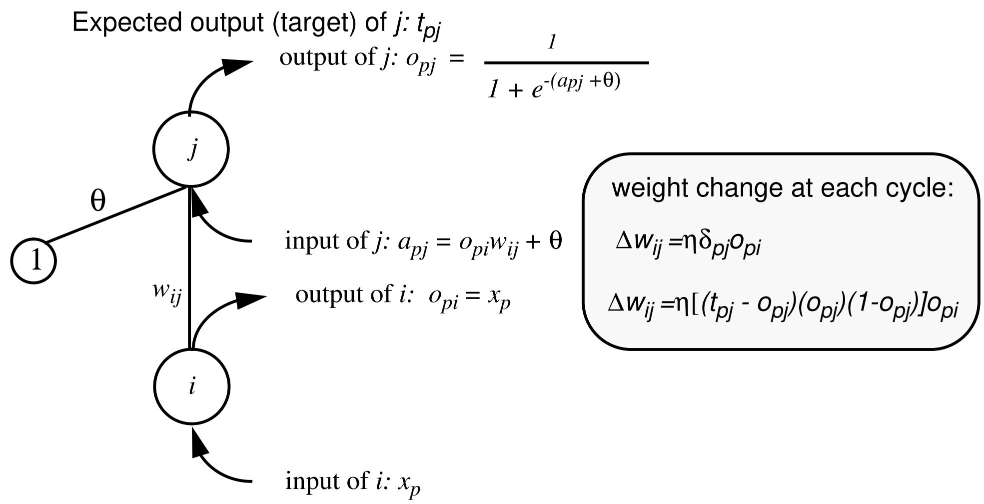


In this one-dimensional example, the derivative of the error function is negative, which indicates that the change in weight should be positive. Note that there may exist local minima in the error surface, so that changing the error according to the derivative may not always result in a global optimum solution.

The basic idea underlying the learning algorithms usually utilized in neural networks is simple. The model starts with small random real numbers as the starting weights. At each training cycle, the error is calculated, and the weights are changed in the direction that minimizes the error. The error surface has as many dimensions as the number of weights, and all the weights obey this basic principle. Gradient descent is a greedy algorithm: it makes the choices that look best at the moment [Cormen, 1990], so it may lead to local minima. In order to reduce the chances of choosing a local minimum as the solution, researchers in neural networks usually develop their models using different random starting weights and select the model with the best prediction capability.

The reader with no interest in specific aspects of neural network learning may skip the end of this subsection and resume reading at Section 2.1.4.

Figure 2.9. Learning in neural networks.



In this example of learning when the logistic activation function is used, the change in weights is proportional to the derivative of the error function, and gradient descent is used. Note that the threshold, or bias θ, can be modeled as a unit which always has activation 1, connecting to the output unit through weight θ. This weight θ is also learned using the delta rule.

As shown in Figure 2.9, in order to perform gradient descent, each weight change between units i and j, or Δw_{ij}, corresponding to a given pattern p, is calculated by the formula

$$\Delta_p w_{ij} = \frac{\partial E}{\partial w} = \eta\delta_{pj}o_{pi}$$

where δ_{pj} is given by

$$\delta_{pj} = (t_{pj} - o_{pj})(o_{pj})(1 - o_{pj})$$

when the logistic activation function is used. The target, or desired output, for a given pattern p in output unit j is represented by t_{pj}, and the output calculated by the network is represented by o_{pj}. The derivation of these formulas is described by Rumelhart [1986].

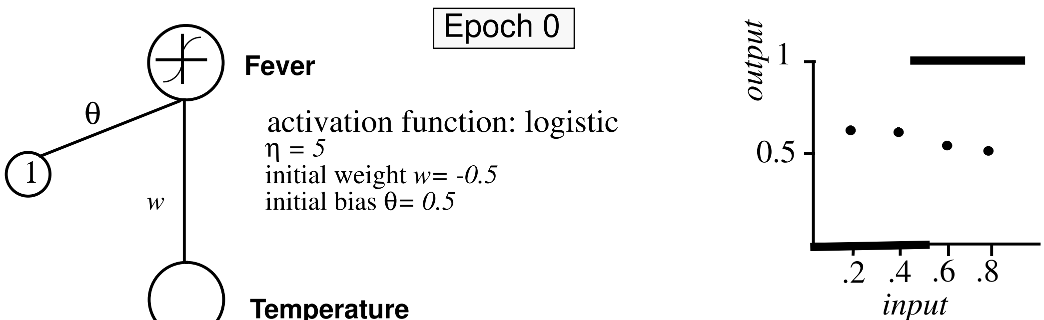
Let us now look at one example of such learning in action. Suppose that a neural network has the trivial problem of deciding which patients have fever, given the data on their temperatures. The training data is shown in Table 1.1.

Table 1.1. Training data for a simple example.

patient	temperature (rescaled)	fever
p1	36.5 (0.4)	0
p2	38.5 (0.8)	1
p3	35.5 (0.2)	0
p4	37.5 (0.6)	1

The neural network for this example is shown in Figure 2.10.

Figure 2.10. Simple network to diagnose fever: Initial weights.



output: $o = \dfrac{1}{1 + e^{-(aw + \theta)}}$

patient	input (a)	output(o)
p1	0.4	0.57444
p2	0.8	0.52497
p3	0.2	0.59868
p4	0.6	0.54983

sq error: 1.11

This example shows how weights are learned. The logistic activation function is used at the output unit, and the initial parameter values are displayed in the figure.

After a first pass, the weight has to be updated according to[1]

$\Delta w = \Sigma \Delta_k w,$

where

$\Delta_k w = \eta \delta_k \, a_k.$

The letter k represents the patient number (e.g., *p1*, *p2*, etc.) and a represents the input.

[1] Subscripts for w and δ denoting the units (e.g. w_{ij}) were ignored in order to simplify the example. They were unnecessary because the example deals with *one* weight, *one* bias, *one* input unit, and *one* output unit only.

Therefore,

$\Delta_{p1}w = (5)\ [(0 - 0.57444)\ (0.57444)(1 - 0.57444)]\ (0.4) = -0.28085$

$\Delta_{p2}w = (5)\ [(1 - 0.52497)\ (0.52497)(1 - 0.52497)]\ (0.8) = 0.47384$

$\Delta_{p3}w = (5)\ [(0 - 0.59868)\ (0.59868)(1 - 0.59868)]\ (0.2) = -0.14384$

and

$\Delta_{p4}w = (5)\ [(1 - 0.54983)\ (0.54983)(1 - 0.54983)]\ (0.6) = 0.33427.$

The total update for w must then be

$(-0.28085 + 0.47384 + -0.14384 + 0.33427) = 0.38342.$

The update for the bias must be

$\Delta_{p1}\theta = (5)\ [(0 - 0.57444)\ (0.57444)(1 - 0.57444)]\ (1) = -0.70213$

$\Delta_{p2}\theta = (5)\ [(1 - 0.52497)\ (0.52497)(1 - 0.52497)]\ (1) = 0.59230$

$\Delta_{p3}\theta = (5)\ [(0 - 0.59868)\ (0.59868)(1 - 0.59868)]\ (1) = -0.71920$

and

$\Delta_{p4}\theta = (5)\ [(1 - 0.54983)\ (0.54983)(1 - 0.54983)]\ (1) = 0.55712.$

The total update for θ must then be

$(-0.70213 + 0.59230 + -0.71920 + 0.55712) = -0.27190.$

The new weight w is then

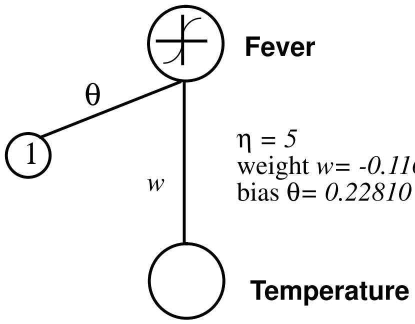
$-0.50 + 0.38342 = -0.11658$

and the new bias θ is

$0.5 - 0.27190 = 0.22810.$

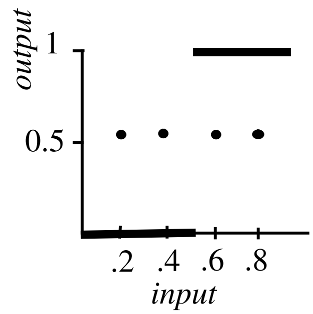
Figure 2.11 shows the network after one epoch.

Figure 2.11. Simple network to diagnose fever, after one epoch.



Epoch 1

$\eta = 5$
weight $w = -0.11658$
bias $\theta = 0.22810$



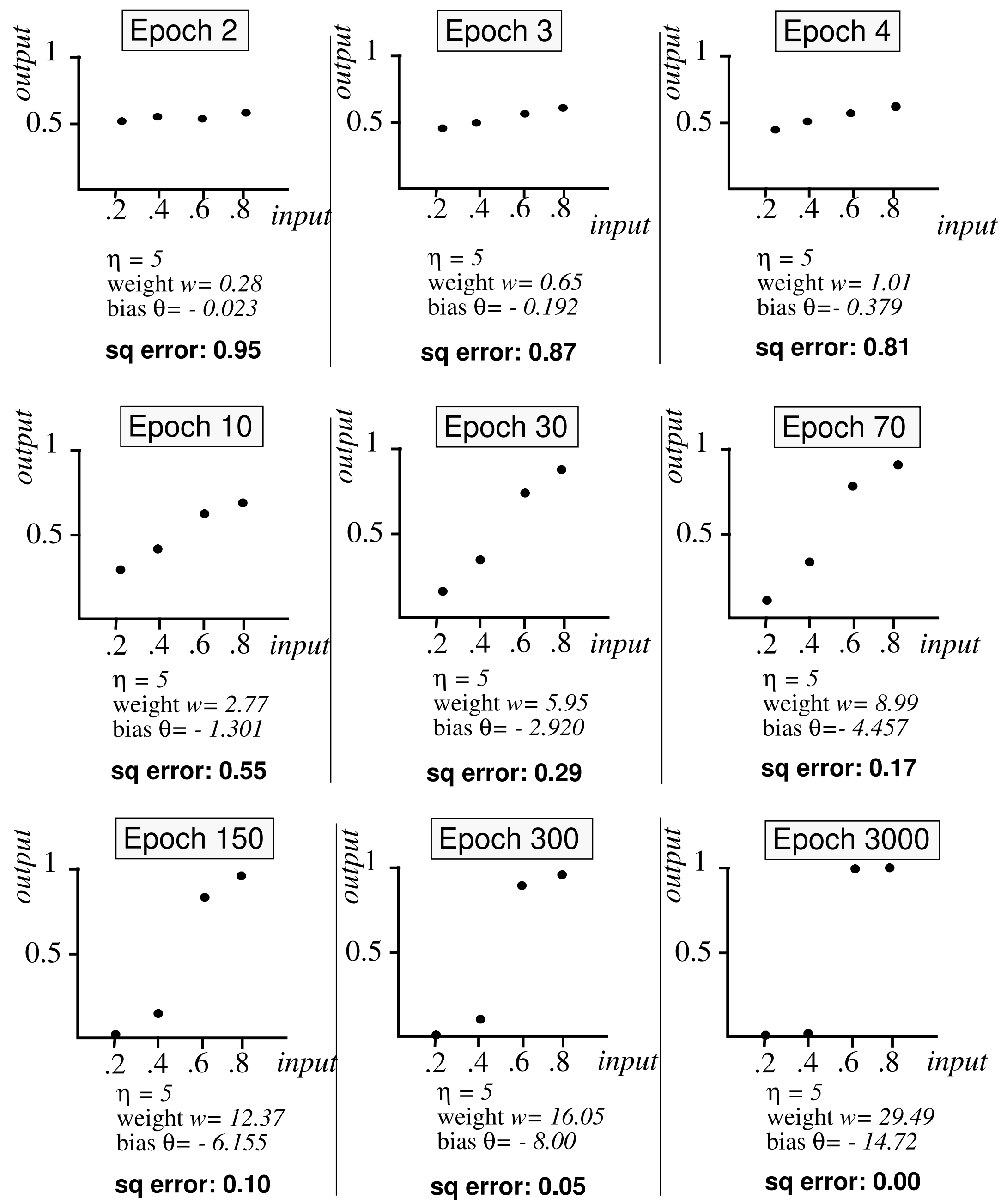
output: $o = \dfrac{1}{1 + e^{-(aw + \theta)}}$

patient	input (a)	output(o)
p1	0.4	0.54524
p2	0.8	0.53365
p3	0.2	0.55103
p4	0.6	0.53945
		sq error: 1.03

The weight *w* was increased and the bias θ was decreased in this first cycle of learning. The output is still wrong.

Figure 2.12 shows what happens to the outputs after several epochs.

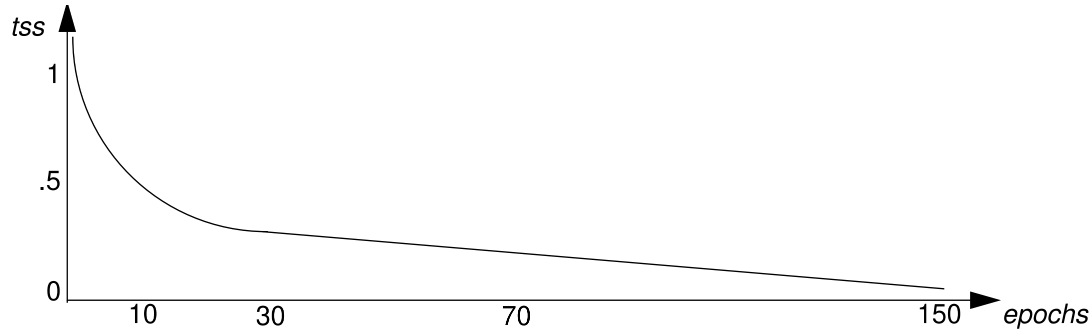
Figure 2.12. Simple network to diagnose fever, after several epochs.



The input is the value for the temperature, and the binary output represents the presence or absence of fever (“0” and “1,” respectively). After relatively few epochs, the neural network determines values for its weights that accommodate all training patterns. As we can see in this example, the training phase may take several cycles until the network perfectly classifies all cases.

Figure 2.13 shows the decrease of the total sum of squared errors (*tss*).

Figure 2.13. Decrease of *tss* and the number of epochs.



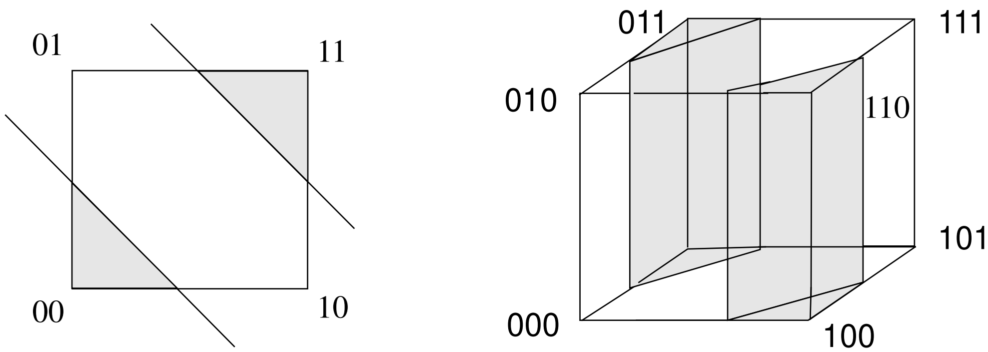
The decrease in error is fast in the first epochs.

In neural network learning, the updates can be done each time a pattern is presented (*training by pattern*), or after all patterns are presented in a training cycle, also called an *epoch* (*training by epoch*). When training is done by pattern, the order of pattern presentation to the network may change the results.

2.1.4 Linear separability

A problem is linearly separable if one *(n-1)* dimensional plane can separate different categories in a space of *n* dimensions [Peretto, 1992]. Figure 2.14 shows the classic XOR problem, in which the two-dimensional space cannot be divided in two by a single line that separates patterns “00” and “11” from patterns “01” and “10.” The same figure shows linear separability failure in a three-dimensional problem where patterns “000,” “010,” “101,” and “111” must be separated from the other patterns. At least two planes are necessary. The convergence theorem states that the perceptron can learn any function that is linearly separable [Rosenblatt, 1962]. Nevertheless, functions that are not linearly separable—which are not uncommon in medicine—cannot be solved by this model. Minsky showed this deficiency with the XOR function [Minsky, 1969], but also noted that, if an additional layer of neurons was added and a function *other than the linear function* was used for activation (since the use of linear activation functions makes multilayered neural networks equivalent to single-layered neural networks), nonlinear problems could be solved. Other authors showed that any function, including nonlinear functions, could be estimated if sufficient nodes were added to this intermediate, or hidden, layer [Hornik, 1989].

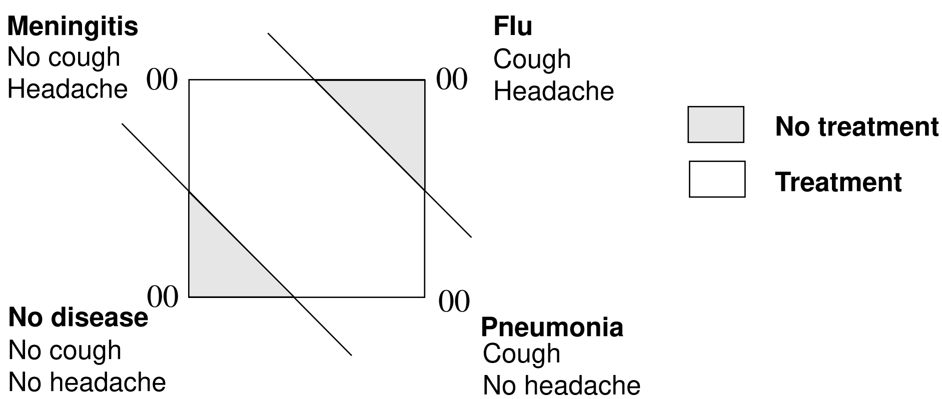
Figure 2.14. Linear separability failure.



The first figure corresponds to the classic "exclusive or" (XOR) problem. It is not possible to use one line to separate patterns "00" and "11" from "01" and "10" in the first figure. In the second figure, linear separability fails in a three-dimensional problem: it is not possible to use just one plane to separate patterns in the shaded corners.

Figure 2.15 shows a hypothetical example where linear separation is not possible.

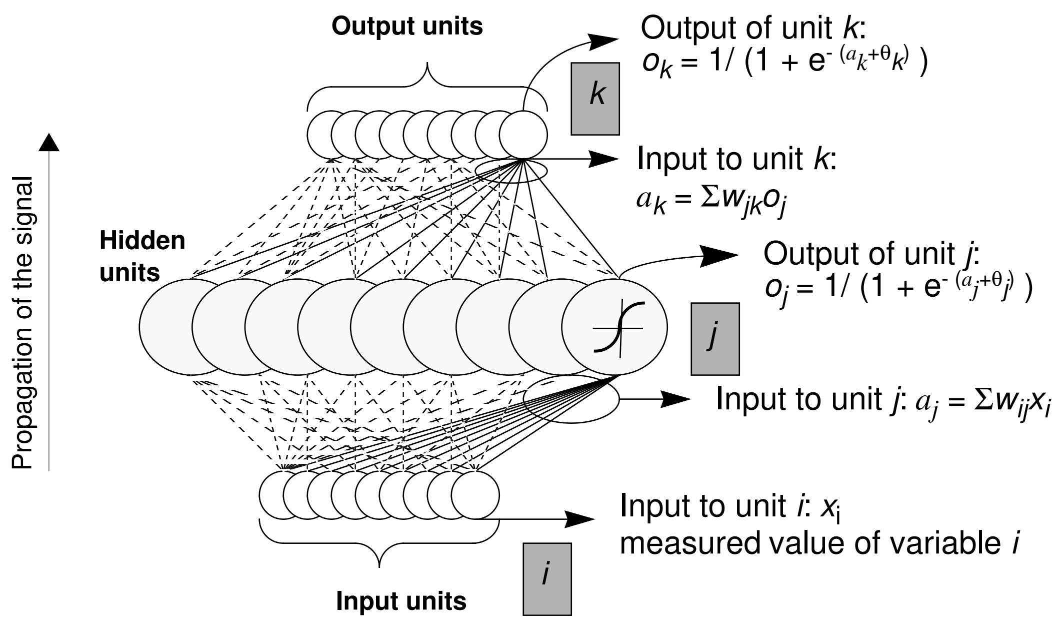
Figure 2.15. Classifying diseases according to treatment.



Suppose the symptoms Cough and Headache determined whether the patient had *No disease*, *Pneumonia*, *Meningitis*, or *Flu*, as shown in the figure. The problem of distinguishing patients who should receive treatment (e.g., antibiotic therapy for meningitis or pneumonia) from those who should not is not linearly separable.

Figure 2.16 shows the basic components of a feedforward neural network.[2] Input values are multiplied by weights that are adjusted iteratively every time a set of patterns is presented. The results of the multiplication are passed through an activation function in each hidden unit of the intermediate layer of nodes (in our figure, the activation function is the logistic). The activation values for the units of the hidden layer will then be multiplied by the weights of the second layer, and the results of these operations will subsequently pass through the activation function of the output layer, providing the final solution.

Figure 2.16. Fundamental elements of neural networks.



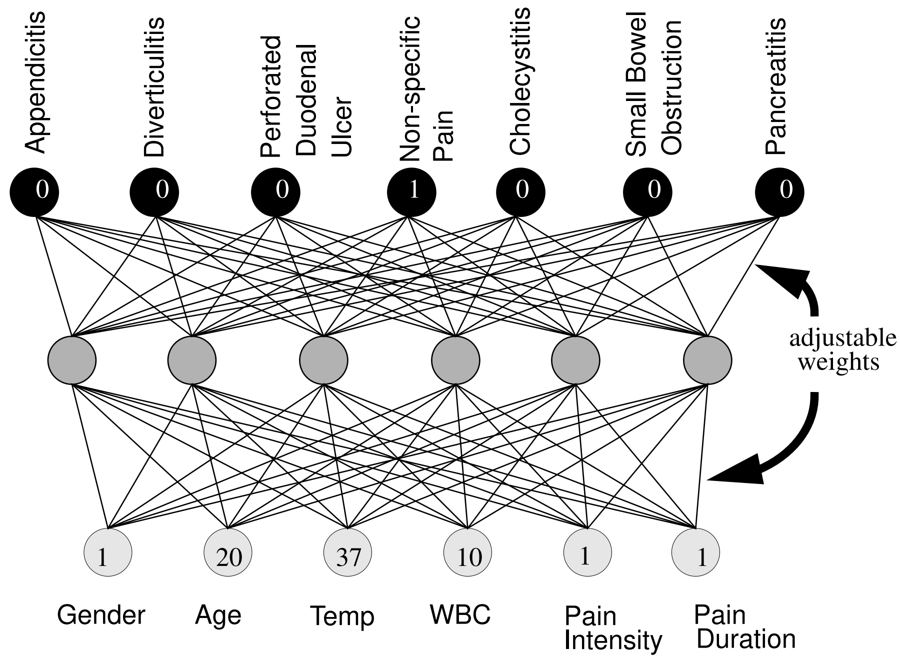
A hidden layer of neurons is added to the perceptron to model complex functions. The hidden layer also has an activation function (the logistic in this example), and receives inputs from the multiplication of input values and weights in the first layer. The input is passed through the logistic activation function, and will be multiplied by weights in the second layer to produce inputs for the output layer.

Note that the output of a multilayered neural network is the composite result of a number of logistic functions. In the case of a simple logistic regression model, there is only one logistic function being used, and only parameters referring to this single function have

[2] Neural networks whose weights are not bidirectional.

to be estimated. In the neural network case, there are several logistic functions, and the parameters for all of them are being estimated. A hypothetical example of a multilayered perceptron is shown in Figure 2.17. A review of existing applications of neural networks in medical basic and clinical sciences is presented in Section 2.2.

Figure 2.17. Multilayered neural network for diagnosing abdominal pain.



In this hypothetical example, a layer of hidden nodes was added to the simpler neural network shown in Figure 2.5. Weights are adjusted by the backpropagation algorithm.

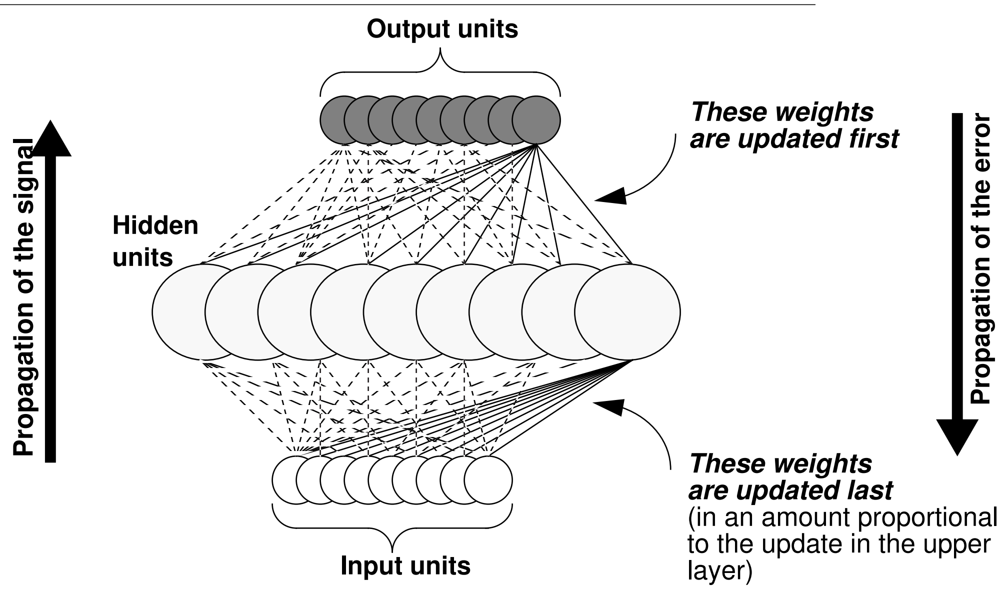
Researchers have known for a long time that multilayered neural networks composed of nonlinear units were able to solve nonlinearly separable problems. The main difficulty, however, was to find an appropriate algorithm to estimate the weights, or a way to *train* the network to *learn* those functions. Although the target for the output units is well defined, so that the delta rule can be used for this layer, the target for the hidden units is not, so that determining how the update should be done in a principled way at intermediate layers was the major obstacle to applying the delta rule. This problem was solved by the backpropagation algorithm, a generalization of the delta rule for multilayered neural networks [Rumelhart, 1986]. Since the publication of the backpropagation algorithm, numerous applications in different fields have used multilayered neural networks successfully. Learning by error backpropagation consists of a modification of the delta rule, applied to all layers.

The next subsection should be skipped by readers who are not interested in specific aspects of the backpropagation algorithm. Reading should resume in Section 2.1.6.

2.1.5 Backpropagation of errors

The backpropagation algorithm consists of the propagation of errors beginning at the output layer, through the hidden layer, and so on, to the input layer, in a backward direction. The weights are therefore updated at each layer, beginning at the output layer. The changes in weights are proportional to the derivative of the errors with respect to the incoming weights. Figure 2.18 shows the propagation of the signals in feedforward neural networks, and the backpropagation of errors.

Figure 2.18. Direction of propagation: signal and error.



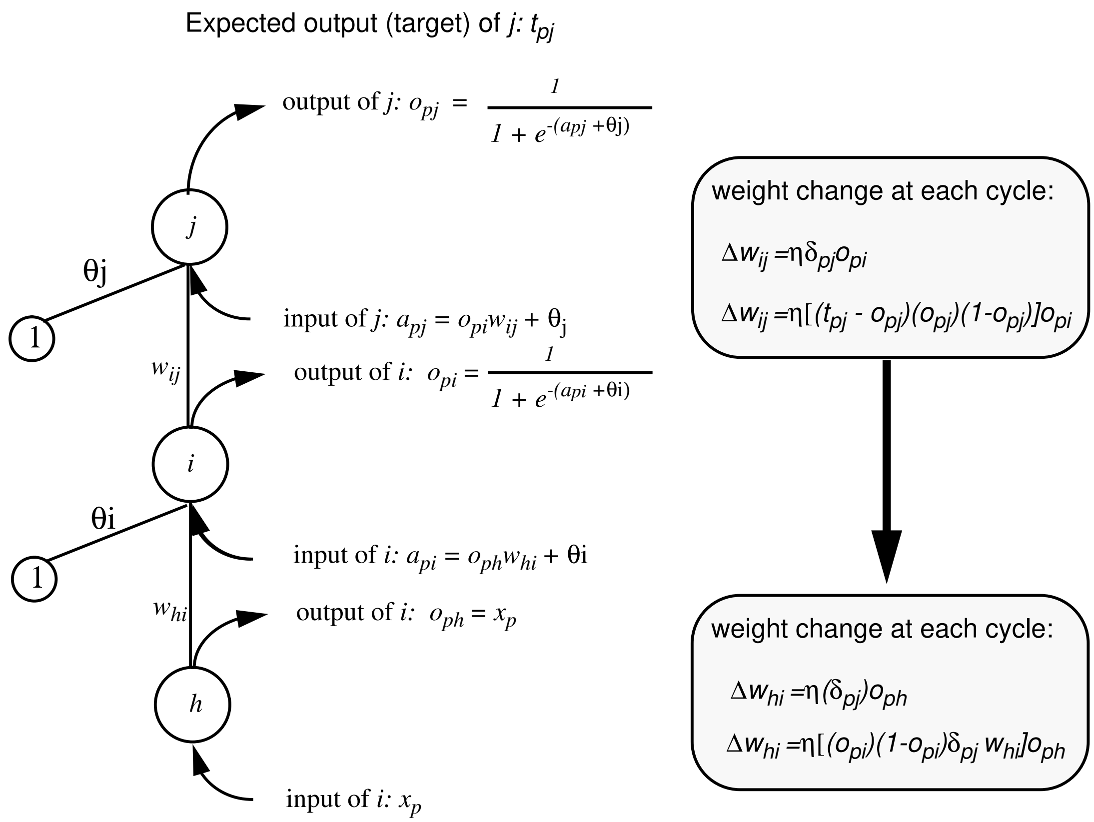
Backpropagation uses the delta rule recursively. Errors are calculated for the output layer, and are then backpropagated to the intermediate layer, in a direction opposed to that of the impulse. Weights are updated according to the errors.

In the backpropagation algorithm, if the unit is in the output layer, its incoming weights are updated as

$$\Delta w_{ij} = \eta \delta_{pj} o_{pi}$$

where η is the learning rate, δ_{pj} is the change in weight connecting unit i with output unit j required by pattern p, and o_{pi} is the output of unit i for a pattern p, as shown in Figure 2.19.

Figure 2.19. Backpropagation in a simple neural network.



In learning by backpropagation, the weights w_{ij} connecting to units in the output layer are modified according to the standard delta rule. The weights w_{hi} in other layers are modified recursively according to these updates. The threshold, or bias θ_j, in the last layer can be modeled as a unit that always has activation 1, connecting to the output unit through weight θ_j. This weight θ_j is also learned using the delta rule. The threshold θ_i in the previous layer is modified recursively according to the update in the last layer.

The delta for a connection between an output unit j and a hidden unit i is the same as the one defined before for a neural network with no hidden units:

$$\delta_{pj} = (t_{pj} - o_{pj})\,(o_{pj})\,(1 - o_{pj})\,.$$

The weight update is therefore

$$\Delta w_{ij} = \eta\,[\,(t_{pj} - o_{pj})\,(o_{pj})\,(1 - o_{pj})\,]\,o_{pi}\,.$$

If the unit *i* is in the hidden layer and it has only one connection to an output unit *j*, its incoming weight connecting to input unit *h* is updated in an amount proportional to δ_{pj}:

$$\delta_{pi} = (o_{pj})\,(1 - o_{pj})\,\delta_{pj} w_{hi}\,.$$

Therefore,

$$\Delta w_{hi} = \eta\,(o_{pi})\,(1 - o_{pi})\,\delta_{pj} w_{hi} o_{ph}\,.$$

The error is calculated at the output layer and its δ_j is defined. The connections leading to the output layer are updated. Next, the connections between the hidden layer and the output layer are updated, using the result obtained previously for the δ_j in the output layer. The weight update function in intermediate layers is, therefore, defined recursively as a function of the update in the next layer, up to the output layer whose target is known.

The derivation of the backpropagation algorithm requires the activation function to be semi-linear: a continuous, nondecreasing, and differentiable function. The backpropagation algorithm applies a steepest descent (or hill-climbing) method to minimize the error function, and therefore it inherits steepest descent's well-known problems: the existence of local minima, the possibility of having multiple solutions, and the difficulty of assuring that the solution found is optimal. Nevertheless, none of the limitations mentioned above has prevented backpropagation-based neural networks from being useful in a variety of real-world settings. Some authors have proposed a system of voting networks in which the same architecture is initialized with different random weights a number of times, and the results are accumulated, in order to maximize the chances of finding the optimal solution [Benediktsson, 1993; Kammerer, 1990].

2.1.6 Interpretation of neural network results

Neural network qualities such as resilience to noise (graceful degradation), local processing, and distributed representation make them appealing as a physiologically plausible model. However, it is exactly these qualities that make them cognitively challenging, imposing obstacles on their prompt interpretation. Explanation has been the stumbling block of neural network models. Even when high accuracy is achieved in neural network models, it is difficult to explain which inputs were taken into account to calculate the outputs, and to provide any insight on how the variables interact. The problem of selecting inputs has been addressed using weight-decay methods that automatically prune out weights whose values are too small, leading sometimes to deletion of nodes [Weigend, 1991], or by preprocessing of inputs by unsupervised statistical methods, such as principal components analysis. Another approach involves the stepwise addition or removal of variables guided by the differences in classification or prediction performance [Baxt, 1992] as it is done in regression models. A general solution to the problem has yet to be achieved. A way of counterbalancing the difficult interpretability of neural network models, while using backpropagation learning, is to impose some structure on the neural network architecture, such as intermediate abstractions [Ohno-Machado, 1994].

2.1.7 Supervised learning

The neural networks presented so far perform supervised learning—that is, they are taught to learn a function of the input values in order to produce an output that is known. In the training phase, the networks receive a number of training examples and learn a function from those examples. The function can be used as new cases are presented. Supervised learning is to neural networks what parameter estimation is to statistical models such as regression, linear discriminant analysis, linear recursive partitioning, and many other non-exploratory data models: It is a way of constructing the model, given the available data. The type of partitions of the data set that each of the above methods allows is different. Whereas in linear methods the partitions of an *n*-ary feature space are done by hyperplanes of *n-1* dimensions, in non-linear methods the partitions may be done by non-linear figures. The assumptions required by each method are also diverse (e.g., in linear discriminant analysis—a parametric model—the categories must have the same Gaussian distribution, varying only in their means, whereas in nonparametric models, such as neural networks or regression trees, no assumption regarding the distribution of the categories is necessary).

Supervised neural networks started to receive attention after the work of Hopfield [1984], who applied principles of statistical physics to the development of their models. The Hopfield model consists of an associative memory that uses the Hebbian rule (see Section 2.1.1) asynchronously to update weights that connect binary units. Weights are updated such that each new pattern that is presented to the network is “attracted” to the one stored pattern that is most similar to itself. Learning in the Hopfield model is based on the minimization of an energy function (also called a Lyapunov, Hamiltonian, or cost function). If the units in a Hopfield model are stochastic, then simulated annealing can be applied in order to decrease the problems with local minima.[3] Variations of supervised learning algorithms may address specific situations. Reinforcement learning is a type of supervised learning in which the network is only told if it produced the correct answer or not. The network receives no information about how much or in what direction weights should be updated. There are a number of supervised learning algorithms and architectures for neural networks the discussion of which is beyond the scope of this work.

Recurrent neural networks are still another type of network in which supervised learning can take place. They can have the same architecture as standard feedforward neural networks, except that they allow feedback connections. Time series have been modeled by this type of networks. An alternative for building recurrent neural networks is the use of backpropagation through time [Rumelhart, 1986].

2.1.8 Unsupervised learning

Unsupervised models for parameter estimation (training) in neural networks have also

[3] Simulated annealing is a phenomenon that occurs in physics. If the temperature of a material is decreased gradually, lower energy states can be achieved than if the temperature is decreased abruptly.

been developed. They are analogous to the statistical exploratory methods of clustering, such as hierarchical clustering or multidimensional scaling, since there are no pre-established results. This characteristic makes evaluations of these models extremely difficult. There is no "gold standard" with which unsupervised models of neural networks can be compared. The advantage of not having to classify the examples that are used to train the network *a priori* is counterbalanced by the fact that the clusters have to be identified and labeled *a posteriori*.

Competitive learning neural networks have been developed by Rumelhart [1986]. Among other neural network models of unsupervised learning for use in a variety of tasks, Kohonen's self-organizing feature maps are popular, having several applications in image processing and other pattern recognition situations [Kohonen, 1982]. Adaptive Resonance Theory has been developed by Carpenter and Grossberg [1988] as another type of unsupervised neural network model, which was further enhanced by the incorporation of principles of fuzzy logic. There are additionally models that combine unsupervised and supervised learning.

2.1.9 Hybrid models

The term *hybrid models* has been used indiscriminately in the neural network literature. While some authors use it to describe models in which both unsupervised and supervised neural network concepts are used in different parts of a connectionist model, others use it to describe models where neural networks are combined with rule-based, classic statistical, and other types of modeling approaches. Examples of the first use can be found in Hertz [1991], whereas Gallant [1988] and Medsker [1994] prefer the second use.

2.1.10 Hardware implementations

Analogous to the development of computerized tomography in the late 1970s, where the understanding and testing of the pioneering ideas were basically developed in software, and the final commercial products implemented the concepts in hardware in order to improve performance, neural network VLSI chips have been continuously produced and enhanced by the hardware industry since the mid-1980s [Mead, 1987]. These processors are based on a parallel distributed processing architecture, and have error and update functions implemented as hardware. Although the advantages in terms of speed are coupled with a certain degree of inflexibility, hardware implementations of neural networks have the potential of being embedded in certain commercial medical appliances.

2.2 Neural Networks as Statistical Tools for Medical Research

A simple search in MEDLINE for articles about computer-based artificial neural networks for the years 1982 to 1994 results in more than 600 citations over the last decade. Other bibliographic data sets also contain numerous publications that deal with the use of neural networks in the health sciences. Several applications of neural network models in medicine deal with the use of artificial neural networks that simulate real neurons, but others use neural networks as a statistical tool for performing classification for diagnosis and prognosis, usually replacing regression models. Applications in the basic sciences, although dominated by neurophysiologic models, contain a significant number of models in molecular biology, where neural networks have been accepted as a useful tool to predict secondary and tertiary structures of proteins and other biologically interesting sequences. The increase in popularity of these parallel processing models has been accompanied by a diversification of areas of applications as well. Whereas in the mid-1980s most of the applications reflected the use of neural networks in the neurosciences, their use in clinical applications has increased considerably. Hardly any specialty in medicine lacks an application of neural network models.

2.2.1 Neural networks versus regression models

Researchers in the medical sciences are familiar with conventional statistical methods for classification, such as multiple nonlinear regression, and linear discriminant analysis.

Neural networks can be used to perform the same tasks, and have both advantages and disadvantages over regression methods (I will consider linear discriminant analysis a special case of regression from now on).

Advantages. Nonlinear regression often involves an educated guess about the degree of the polynomial function whose parameters are being estimated and the forms of interactions in which the independent variables may relate to each other. Suppose that there are two independent variables and one dependent variable. The independent variables may appear in several terms for each degree of the polynomial function. For example, if the variables are x, a possible regression model with degree 2 would be:

$y = \mathrm{a}x^2z^2 + \mathrm{b}x^2z + \mathrm{c}x^2 + \mathrm{d}xz^2 + \mathrm{e}z^2 + \mathrm{f}xz + gx + \mathrm{h}z + \mathrm{i}$.

As the number of independent variables increases, the number of possible regression models becomes intractable. Furthermore, parameter estimation requires operations on matrices, which are often of limited size in commercial software packages. In neural networks, it is not necessary to specify the degree of the polynomial in advance, or the interactions between variables. Parameter estimation is a simple process that requires only repeated (though sometimes time-consuming) additions and multiplications of real numbers. If a sufficient number of hidden units is present, functions of any complexity can be approximated by neural networks [Hornik, 1989].

Other types of nonlinear regression, such as project pursuit regression, generalized additive models, and multivariate adaptive regression splines, have also been approximated by neural networks [Cheng, 1994]. A full account of the performance and estimation time trade-offs for different types of models is still necessary.

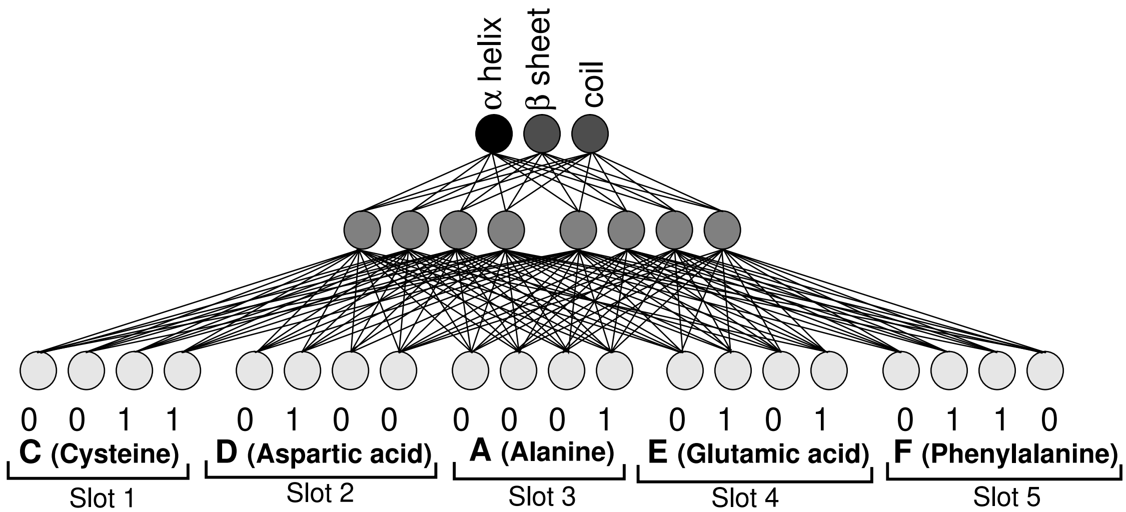
Disadvantages. Linear models are easily constructed in conventional linear regression and mechanisms for comparing the performance of these models have been well studied. If the function being approximated is linear, training in neural network models will be very slow and offers no advantage over the calculation of parameters by Fisher's linear discriminant or other conventional statistical methods. In complex examples, however, it is not always evident whether the function being approximated is linear. In neural network models, there are no coefficients that can be interpreted, as there are in regression models. Therefore, there is no indication as to which inputs (or independent variables) have a stronger influence on the results of the classification, especially when there are interactions between variables. The interpretability of neural networks is one of their most criticized features. Currently, no way of interpreting neural network weights has been universally accepted.

2.2.2 Applications in the basic sciences

Although the majority of neural network applications in the basic sciences are related to simulations of connections of real neurons to reach the ultimate goal of understanding physiological systems (a biological approach), I will focus on the use of artificial neural networks as a tool for performing classification (a statistical approach). Among other applications, neural networks have been used to identify pathogens in microbiology [Freeman, 1994] and to design and discover new drugs in pharmacology [Weinstein, 1994]. Their most frequent use, however, has been in the analysis of sequential data in biological structures. The Genome Project has made possible the storage and fast retrieval of a variety of sequences.[4] The analysis of these sequences has been the focus of many researchers, who apply classic and Bayesian statistics, linear-programming techniques, and also neural network models to classify and compare structures [Presnell, 1993].

[4] The Genome Project is an initiative of the U.S. government to promote the development of molecular biology and the understanding of the human genome. Several institutions across the country are engaged in massive sequencing of the human genome and storage of this information in centralized databases.

Figure 2.20. Prediction of the secondary structure of proteins.



A window of five slots of aminoacids. Each amino acid is represented by four inputs. For example, the aminoacid *Cysteine* (also known by the letter "C," the third letter of the alphabet), is represented by the binary number 3. Output categories are secondary structures: α helix, β sheet, and coil.

Neural networks have been applied to the prediction of secondary and tertiary structures of proteins, DNA, and other biologically interesting sequences. Several authors use the number of each of the 20 aminoacids, and amino acid properties such as hydrophobicity and charge, to predict the existence of alpha helices, beta sheets, or coils. Others incorporate proximity information, by choosing a "window" of aminoacids that may fall into one of the spatial categories. The representation of amino acids can be localized, using 20 nodes for input for each slot in the window, or distributed, using a four nodes, as shown in Figure 2.20.

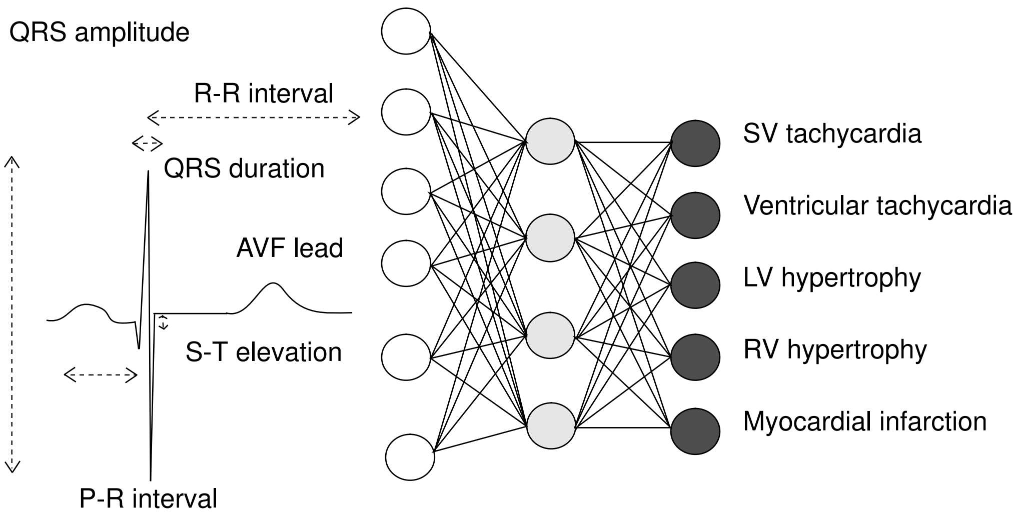
2.2.3 Applications in clinical medicine

Classification, or pattern recognition, is one of the most common uses of neural networks in medicine. Statistical methods for classification in health sciences have been used in medicine [Lew, 1983] to solve problems as different as (1) prediction of diagnoses, (2) prediction of outcomes, such as length of stay, charges, prognoses, and rate of complications [Knaus, 1991], (3) determination of cut-off values for diagnostic tests, and (4) determination of risk or disease profiles [Kannel, 1993]. The medical literature also has numerous examples of neural network models. They have been applied as statistical tools to solve problems including (1) prediction of diagnoses, such as myocardial infarction [Baxt, 1991], giant cell arteritis [Astion, 1994], several types of cancer [Maclin, 1994 and 1992; Rogers, 1994; Wilding, 1994]; (2) prognoses, such as valve-related complications in heart disease [Katz, 1994 and 1993], length of stay in the intensive care unit [Doig, 1993; Tu, 1993; Buchman, 1994], admission to the psychiatry ward [Somoza, 1993], outcomes of liver transplantation [Doyle, 1994], outcomes of oncologic treatment [Burke, 1994; Kappen, 1993; Ravdin, 1992; McGuire, 1992], failure to survive following cardio-pulmonary resuscitation [Ebell, 1993], and survival after trauma [McGonigal, 1993]; (3) interpretation of diagnostic tests, such as pancreatic enzymes [Kazmierczak, 1993], thyroid panels [Sharpe, 1993; Bolinger, 1991; Ohno-Machado, 1994], and tumor markers; and (4) decision support, such as assessment of the adequacy of weaning patients from ventilators [Ashutosh, 1992] and of esophageal intubation [Leon, 1994]. The overwhelming majority of these articles involve applications of the backpropagation algorithm.

2.2.4 Applications in signal processing and interpretation

Neural networks have been applied to the study of ECGs [Bortolan, 1993; Edenbrandt, 1993], EEGs [Kloppel, 1994], EMGs [Chiou, 1994] and hemodynamic signals [Laursen, 1994]. In ECG analysis, both supervised and unsupervised neural networks have been used to assist the diagnosis of myocardial infarction or ischemia [Heden, 1994], arrhythmia [Evans, 1994; Griffin, 1994; Yang, 1993], and left ventricular strain [Devine, 1993]. Inputs are usually abstracted features, but some authors have worked with raw digital signals. McAuliffe [1993] used neural networks to compress Holter data. Figure 2.21 shows how abstracted features can constitute inputs in a neural network that is able to diagnose cardiac conditions.

Figure 2.21. ECG interpretation.



Inputs for signal processing applications may either be abstracted features, such as P-R interval or S-T elevation, or raw readings from the ECG.

Similar work has been developed for EEG analysis. Neural networks have been used to diagnose Alzheimer's disease and dementia [Pritchard, 1994; Anderer 1994], multiple sclerosis [Wu, 1993], and epilepsy [Jando, 1993]. The scope of applications in signal analysis is broad. It also encompasses the use of EMG signals to drive member prosthesis [Hudgins, 1993] and the analysis of hemodynamic data to detect life-threatening events [Laursen, 1994].

2.2.5 Applications in image processing

As in signal analysis, several applications in image processing utilize the pattern-recognition capability of neural networks. As digital data become more pervasive in radiology, nuclear medicine, and even in other medical areas in which images are fundamental tools, such as dermatology, pathology, and endoscopy, computer-based systems for image analysis become increasingly more useful. It is not only for digital data that neural networks are used. As explained above for signal-processing applications, input units in neural network systems may be abstracted features instead of actual pixel values.

Neural networks for X-ray analysis have been applied in the domains of mammography [Zhang, 1994] and chest radiographs [Chiou, 1994; Lin, 1993; Lo, 1993]. Recent applications in ultrasonography include the examination of the gallbladder [Rinast, 1993] and the vascular system [Akay, 1993, Allen, 1993]. Computerized tomography, magnetic resonance, and nuclear medicine imaging have also witnessed an impressive growth in neural network applications over the last few years.

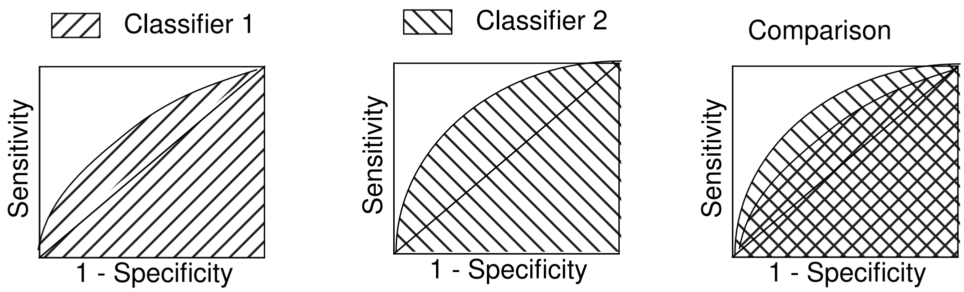
2.3 Evaluating Neural Network Applications in Medicine

The recent emphasis on formal evaluation of neural network models has been responsible for the increasing popularity of neural network models among health care researchers. Earlier systems demonstrated only that neural networks were able to learn patterns in a given (training) set. Researchers did not evaluate how these models would perform in different (or test) sets, did not compare their performance with other types of models, and did not assess how much information was gained by using them.

2.3.1 Neural networks as diagnostic tests

One important issue in diagnostic test evaluation is the balance between Type I errors (when the null hypothesis is rejected, but should have been accepted) and Type II errors (when the null hypothesis is accepted, but should have been rejected). This issue was not addressed by earlier work in the field of neural network classifiers. Penalties for false positives were assumed to be the same as penalties for false negatives. This assumption rarely holds in real-world problems. Systems must be evaluated not only in terms of total accuracy (the percentage of correct classifications), but also on how much information they provide over a simple educated guess that all cases belong to the most frequently represented category (which would result in a 99 percent accuracy in a data set where 99 percent of the cases belong to a given category).

Figure 2.22. ROC curve.



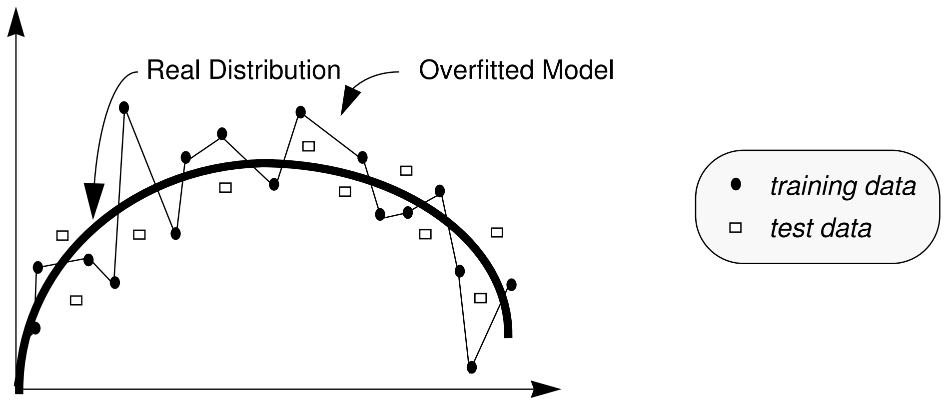
Classifier 2 is better than Classifier 1, since the area under the ROC for Classifier 2 is larger than that for Classifier 1.

Sensitivity and specificity, as well as positive and negative predictive values, must be used to evaluate performance of such systems. Receiving Operating Characteristic (ROC) curves may also be used in order to evaluate performance for all possible thresholds of a given classifier [Swets, 1973]. Figure 2.22 shows the areas under the ROC curves for two different classifiers. The classifier with the largest area is usually considered the best if all other features (such as price, risk, discomfort, availability) are the same.

2.3.2 Avoiding overfitting: Training, test, and validation sets

Several early applications of neural networks in medicine reported the fitness of the model to a given set of data. The impressive results usually were derived from overfitted models, where too many free parameters were allowed. Figure 2.23 shows an example of a perfect fit to the training data that was sampled from a quadratic distribution.

Figure 2.23. Overfitting data.



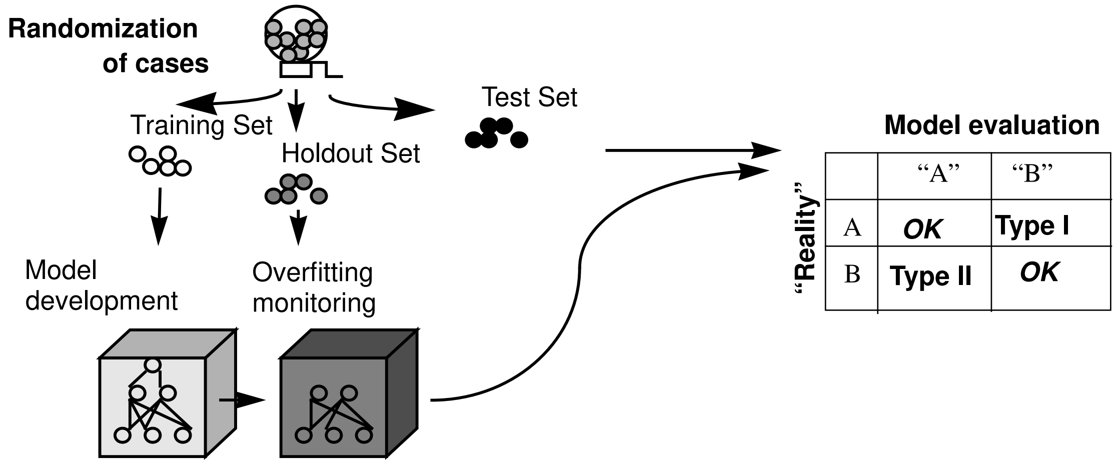
Overfit to training data is caused by allowing too many free parameters (weights and/or cycles) in the neural network model. Overfitted models do not generalize well to new data.

In overfitted models, near-perfection is achieved for modeling a given set of training data. However, since the training set is just a sample of the real population, and even the sampling variations were accounted for in the overfitted model, new data will probably not be well modeled. As mentioned before, multilayered neural networks are known to be able to approximate any function, provided that enough hidden units (and consequently enough weights) are utilized. It is therefore expected that, for a given training set, a neural network system will be able to reach 100 percent accuracy, by simply “memorizing” all the cases.[5]

The grand challenge is, however, to generalize these results for a new set of data not used in the training phase. For this purpose, training, holdout, and test sets may be used, as shown in Figure 2.24.

[5] The most commonly used statistical models do not suffer as much from the problem of overfitting, because they are often limited in the way they can fit the data. For example, the linear regression model requires that data be modeled in a line, which is a strong limitation on how the model can fit all data points. Therefore, R^2 values, which are often used as a measure of how much the data can be explained by the model, are relatively good measures of generalization capability for these models. They do not apply to neural networks.

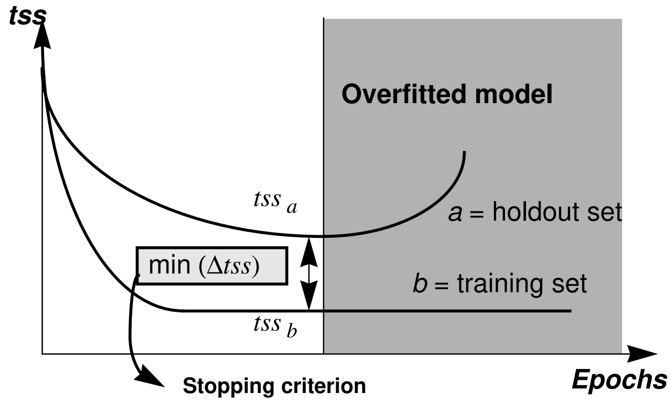
Figure 2.24. Training, holdout, and test sets.



After random assignment of cases, training, holdout, and test sets of data are created. The training set is used to build the neural network model, the holdout set is used to monitor overfitting, and the test set is used to evaluate performance on new cases (generalization).

In neural network models, the error in the holdout set is monitored to determine when the learning phase should be terminated. Both the training and the holdout set errors are high when the system starts learning from the initial random distribution of weights. The error in the training set will always decrease, and it may eventually reach zero when the system is overfitted. The error in the holdout set will decrease in the beginning of the learning phase, but it will at some point begin to increase again. At this point, learning should stop. Figure 2.25 shows the stopping criterion to avoid overfitting in neural network models.

Figure 2.25. Stopping criterion.



In this example, *tss* is the total sum of squared errors. A holdout set is used to determine the stopping criterion for training in a neural network. When the error in the holdout set starts to increase, it is time to stop training to avoid overfitting.

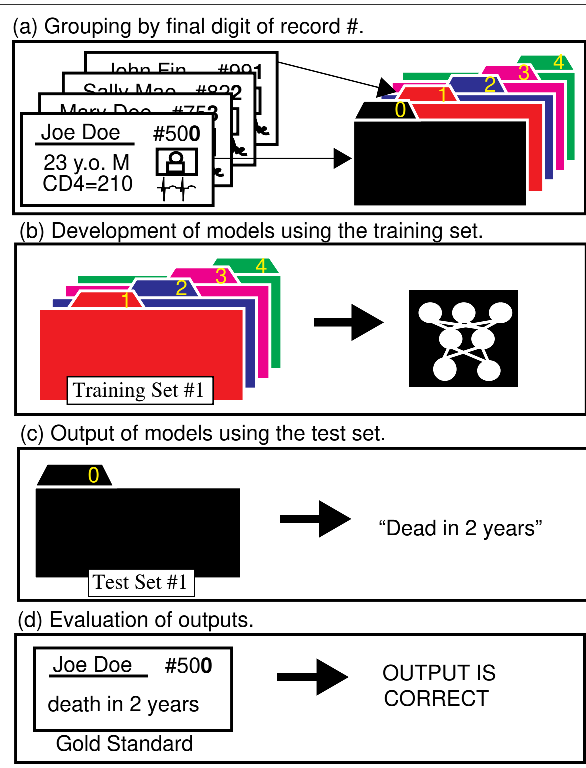
2.3.3 Techniques for dealing with small samples

It is not always possible to divide the sets for training, monitoring overfitting, and testing the networks, because data collected in real life are often not abundant. Neural networks can generalize to new cases only if they are trained on a significant set of data. By dividing a small set into training, holdout, and test sets, the researcher may lose important cases for building an accurate model.

There are two techniques for dealing with small samples that deserve special attention, cross-validation and bootstrap. In both of them, an approximation of the true error is sought, while the full model is trained on all the examples. The apparent error of the training and holdout sets guides the learning phase, but the evaluation is done on an example that has *not* been yet presented to the network, or a test set, so a better approximation of the true error rate is achieved.

Cross-validation: leaving n *out.* In this method, the sample is divided into groups of *n* elements. All groups except one are used to train the network. The group that is left out is used for testing the model, and the results are recorded. A different group is then chosen to be left out, and the network is trained with all other groups, and tested with the one left out. This process is repeated until all groups have been left out once for testing. All results are combined to approximate the true error [Stone, 1977]. In the example of Figure 2.26, we divided a set of patients into 10 groups by systematically classifying them according to the last digit of the database record number. Therefore, all patients with record numbers containing “0” as the last digit were grouped in the first set (Group 0), all patients with record numbers containing “1” as the last digit were grouped in the second set (Group 1), and so on. The training set was composed of those records left behind after each group was selected. For example, when Group 0 was selected as the test set, all other records containing last digits different from “0” would compose the training set and would be used to build the model. This process was repeated 10 times, so that every group was used once as a test set, and all patients were tested.

Figure 2.26. Example of *leave-n-out* method.



In this example, a neural network determines whether a patient will be alive or dead at a given interval.

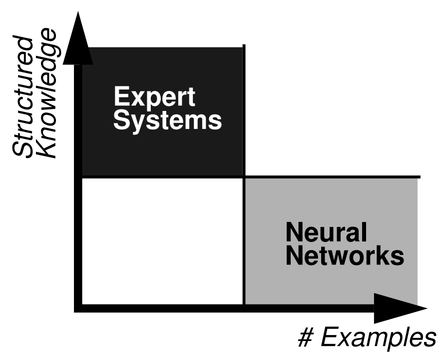
Bootstrap. In this method, the sampling is done differently: Replacement is allowed in the composition of the training set, up to the point where this set contains the same number of cases as the original sample. The test set is composed of the cases that were left out. Several models are constructed and the final evaluation is a weighted average of error in the training and test sets [Efron, 1979]. The procedure for the bootstrap 0.632 estimation method involves the creation of training sets by sampling the original set with replacement, until the number of cases in the training set is the same as the original set. The test set is composed of the cases that did not compose the training set. The procedure is repeated many times, so that several training and test sets are created. Errors for both the training and the test sets are calculated, and the final bootstrap estimate of the total error, which is an approximation of the total error, is an average of the weighted sum of the training and test errors of each pair of training and test sets [Walker, 1992]:

*Bootstrap error = 0.632 * bootstrap test-set error + 0.368 * bootstrap training-set error.*

2.4 Considerations about the appropriateness of neural network models

Neural network models are not a panacea for all problems of data modeling for prediction. Computer-based systems based on production rules have also proven to be capable of exhibiting intelligent behavior in medical tasks, such as diagnosis in the domain of infectious diseases [Shortliffe, 1976] and selection of optimal parameter values for ventilators in ICUs [Fagan, 1980]. In these structured domains, rules can be provided by human experts, and the scrupulous selection and concatenation of these rules produces a model that is able to perform adequately when new cases are presented. In these pioneering systems, the machine-learning phase was substituted by human knowledge acquisition, whereby a knowledge engineer interacts with an expert. Even though later knowledge acquisition tools further facilitated knowledge acquisition, automating part of the process [Musen, 1989], knowledge acquisition is still considered the bottleneck in the development of expert systems, partly because in certain domains structured knowledge is scarce, making it difficult to extract explicit knowledge from experts. If that is the case, but data are abundant, neural network models can be used effectively. Figure 2.27 illustrates this idea. Early work in machine learning addressed the problem of discovering diagnostic rules from a set of diagnosed cases of diseases. In his work, Michalski [1980] claimed that a system based on induction of rules from a case library performed better than a system constructed from decision rules volunteered by experts. A system of neural networks also acquires knowledge from a case library, and has the potential of providing accurate classifications. The main difference between neural network and symbolic approaches to inductive learning[6] is that the former encodes knowledge in its internal weights and not in explicit rules.

Figure 2.27. Expert systems vs. neural networks.



Neural networks and knowledge-based expert systems are not competitors. Knowledge-based expert systems should be used when there is abundance of structured knowledge and experts are available. Neural networks should be used when data are abundant and structured knowledge is not.

Neural networks require that a large number of cases be available for the learning phase. Even though human knowledge is required to determine the architecture of the network and the adequate values for some initial parameters, most of the learning is done by the system itself. Nevertheless, the complexity of these nonlinear models hinders their explanation capability. Recent research in trying to determine which features are important in the final model and how they are related is still in its infancy. General guidelines for

[6] Learning by generalizing specific facts or observations [Michalski, 1980].

when to use neural networks and which architectures to choose from are not yet available.

2.5 Summary

Neural networks, or connectionist systems, are computer-based tools inspired by the vertebrate nervous system that have been increasingly used in the past decade to model biomedical domains . The backpropagation algorithm has been used in the vast majority of applications in the fields of basic sciences, clinical sciences, and signal and image analysis. Pioneer researchers of connectionist systems based the evaluation of these models on the apparent error in a training set. The use of training and test sets to avoid overfitting and to provide an approximation of the true error rate is now the standard. Neural networks should not be used indiscriminately, and they can be more useful if the domain being modeled lacks structured or explicit knowledge, if the number of examples is high, and if explanation capabilities are not essential.

CHAPTER 3

Rare Category Recognition in an Artificial Data Set

This chapter describes the limitations of backpropagation-based neural networks in dealing with rare-category recognition. In Section 3.1, the problem is described and existing solutions are discussed. In Section 3.2, the existence of the problem is demonstrated and quantified in an artificial data set where the classification is deterministic. An example using a similar data set, but with probabilistic classification, is presented in Section 3.3. A current solution to the problem of recognizing infrequent categories is also attempted: patterns are replicated so that all categories are equally represented. Although learning speed is enhanced with replication, the final solution is not correct. The increase in sensitivity for low-frequency categories is accompanied by a decrease in specificity for those categories.

3.1 Rare Categories and Backpropagation-based Neural Networks

The problem of learning rare categories (or rare *classes*) in neural network research has been described by Lowe [1990] and briefly mentioned by Curry [1990]. As Lowe states, "Adaptive networks trained on a 1-from-*c*[1] classifier problem exhibit a strong bias in favour of those classes which have the largest membership in the training data, [which] is an undesirable feature of networks (and many other standard classifiers) in problems where information on one particular class may be more difficult or expensive to obtain than other classes, and where the relative importance of the classes follows a different distribution from their frequency of occurrence." Lowe devised an analytic regularization scheme to compensate for the uneven class membership and tune the network by error weighting. This weighting may be done according to the prior probabilities of each category, and is therefore very similar to the solution discussed later in Section 3.2.4. Curry and Rumelhart [1990] used a similar weighting in their work on classification of chemical compounds, as discussed in Section 3.2.4.

Even though researchers in medical informatics are often looking for low-frequency data or rare categories, the latter are difficult to recognize in certain types of machine-learning methods, including backpropagation-based neural networks. The difficulty is often due to the fact that the error corresponding to frequent categories, being higher than that of rare categories, drives the learning algorithm and slows the learning phase. That is, categories that occur more frequently account for a larger error than categories that are infrequent, as is shown in the error function. The standard error function to be minimized in a backpropagation-based neural network for each output node is usually

$$E = \frac{1}{2}\sum_{i}[\zeta_i - O_i]^2$$

where ζ_i is the expected output for pattern i, and O_i is the output provided by the network [Hertz, 1991]. The changes in weights in the backpropagation algorithm are proportional to the first derivative of the error function. Since the error function is the result of the sum

[1] A classifier that determines that a case belongs to one out of *c* categories.

of squared errors of *all patterns in all categories*, categories with higher frequency will have a stronger influence on the weight changes.[2] Utilities can be taken into account in the process of changing weights if the error function is changed to reflect the researcher's interest in detecting a given category. For example, if it is important to diagnose an infrequent disease, the error corresponding to a failure to recognizing that disease should be increased by a certain factor (utility or importance factor), so that it becomes more easily identifiable by the neural network. Figure 3.1 shows how utilities can be incorporated into the error function. In this case, however, a different network will have to be trained each time the utilities change, and the recognition of a category for a given pattern, which would be analogous to determining the probability of a disease given a set of symptoms, cannot be disambiguated from the process of making an optimal decision using a physician's specific set of values, which would then include issues such as costs, risk, patient's preferences, and so on.

Figure 3.1. Embedding utilities in the error function.

$$E(W) = \frac{1}{2}\sum_{p}\sum_{i}\left(\zeta_i - o_i\right)^2$$

standard error function

"Utilities"

$$E(W) = \frac{1}{2}\sum_{p1}\sum_{i}\left(\zeta_i - o_i\right)^2 I_1 + \frac{1}{2}\sum_{p2}\sum_{i}\left(\zeta_i - o_i\right)^2 I_2$$

modified error function

Utilities can be embedded in the error function by multiplying the error corresponding to category (output node) p_1 by I_1 (importance or utility constant), and the error corresponding to p_2 by I_2.

Traditional classification methods, such as linear discriminant analysis, also have difficulties in detecting infrequent categories [Gray, 1976]. If the variability of the most frequent categories is high, then a rare class may be considered just another instance of the most frequent class, and no discrimination will be possible. On the other hand, if all

[2] The use of the cross-entropy error function, explained in Section 3.3.2, is also affected, though to a smaller degree, by the prior probability of the categories, thereby hindering the learning of rare categories.

classes are equally represented, neural networks should be able to make the distinction easily. Unless neural network applications address the problem of discriminating low-frequency classes, their use in medical applications will not scale up to useful real-world applications.

Curry and Rumelhart [1990] used a "rule of thumb" that dictates that, whenever the frequency of a certain category is below 1 percent, backpropagation neural networks will have difficulties in identifying the rare category. Two solutions for this problem have been proposed: (1) changing the prior probabilities of each category, by replicating patterns in rare categories or removing patterns in frequent categories, such that all categories become equally represented as inputs in a neural network; (2) changing the error function by embedding utilities or import values, such that errors corresponding to infrequent categories have a higher weight than those corresponding to frequent categories. The implications of the use of each method are that (1) information on prior probability of each category is lost, and the artificially replicated examples increase the number of training patterns, slowing learning in the backpropagation algorithm, without providing any additional information, and (2) whenever utilities change, the whole neural network has to be retrained. These implications were briefly discussed in Chapter 1. We will quantify in the next examples the problem of recognizing infrequent categories in two artificial data sets. In the first example, the categories are deterministic, such that equal patterns are always classified the same way. In the second example, categories are determined stochastically, that is, the same pattern will have a certain probability of being labeled "category 1," "category 2," and so on.

3.2 Example I: Deterministic Sorting of Binary Numbers

I will demonstrate the problem of learning infrequent categories in a simple artificial example. Although meaningless from a medical point of view (since medical problems tend to be much more complex), this example illustrates the basic ideas underlying the recognition of infrequent categories and current methods to minimize this problem. Once the reader is convinced of the existence of this problem and the drawbacks of current solutions, it will be easier to switch to real examples, which are provided in Chapter 5.

3.2.1 The data set

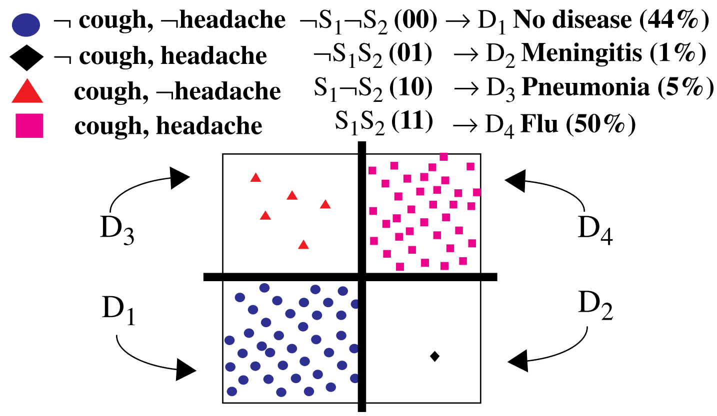
I created an artificial data set using a known distribution. In this data set, four categories (Category 0, Category 1, and so on) have to be discriminated. There were two attributes for each pattern, which constituted the binary representation of the number assigned to each of the classes ("00" was the pattern that corresponded to Category "0," "01" corresponded to Category "1," "10" corresponded to Category "2," and "11" corresponded to Category "3"). Each input unit corresponded to one digit of the binary number. All the units were binary. The input patterns, frequency of each type of pattern, and expected output categories are shown in Table 3.1.

Table 3.1. Distribution of patterns for Example I.

Pattern			Frequency	Output (Disease)	
00	$\neg S_1 \neg S_2$	●	44%	0	D_1
01	$\neg S_1 S_2$	◆	1%	1	D_2
10	$S_1 \neg S_2$	▲	5%	2	D_3
11	$S_1 S_2$	■	50%	3	D_4

Consider that the problem could be formulated as follows. There are two symptoms, S_1 (cough) and S_2 (headache), and four conditions, D_1 (No disease), D_2 (Meningitis), D_3 (Pneumonia), and D_4 (Flu). If the patient has both symptoms S_1 and S_2 (cough and headache), then the diagnosis should be D_4 (Flu); if the patient has symptoms S_1 (cough) and $\neg S_2$ (no headache), the diagnosis should be D_3 (Pneumonia); if the patient has symptom S_2 and $\neg S_1$ (headache, but no cough), the diagnosis should be D_2 (Meningitis); and finally, if the patient has symptoms $\neg S_1$ and $\neg S_2$ (neither cough nor headache), the diagnosis should be D_1 (No disease). Figure 3.2 shows how the patterns are distributed and the perfect classification that is achieved by defining categories in quadrants. The task of a machine-learning method is to define the boundaries of such quadrants.

Figure 3.2. Example I. Perfect classification of patterns in four categories.



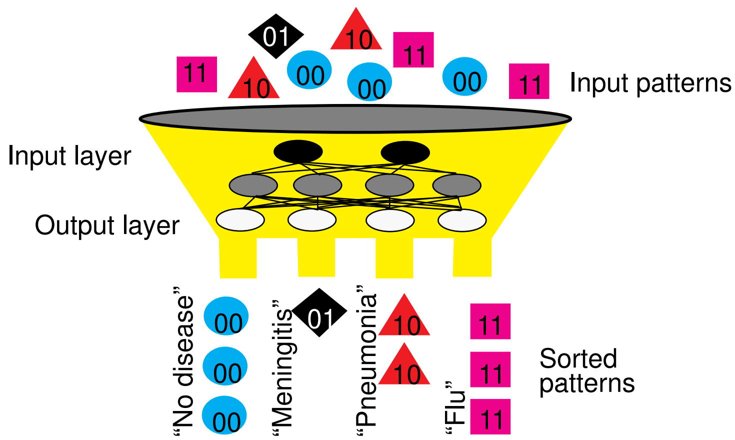
In this simple example, classification of diseases is deterministic: $\neg S_1 \neg S_2$ (00) corresponds to D_1 (No disease), $\neg S_1 S_2$ (01) corresponds to D_2 (Meningitis), $S_1 \neg S_2$ (10) corresponds to D_3 (Pneumonia), and $S_1 S_2$ (11) corresponds to D_4 (Flu).

3.2.2 The neural network classifier

A standard feedforward neural network, with 2 input, 4 hidden, and 4 output units, as displayed in Figure 3.3, was created to learn these categories. The network had 24 weights to be iteratively learned. Using a fixed learning rate of 0.01, and no momentum term,[3] the network took an average of 148,791 epochs to converge to a perfect solution. A perfect solution was defined to be achieved when the activation of the correct output unit was at least twice that of the other output units. No noise was added to the data. Training was done by epochs. I performed 10 simulations for each system, starting with different initial weights. Although the nature of the problem allows a simple perceptron (a one-layered neural network) to converge to a solution, my study focused on the behavior of the back-propagation algorithm for multilayered neural networks. The perceptron's performance on this problem was extremely good, as expected, but it would not be as good in the case of a nonlinearly separable problem, as we will see later in Chapter 5.

[3] The momentum term can be added to the weight update function to avoid weight oscillation [see Rumelhart, 1986].

Figure 3.3. Neural network to sort two-digit binary numbers.

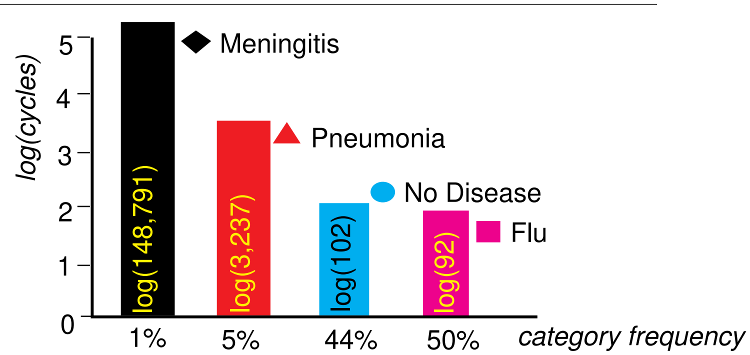


All patterns are sorted by a standard neural network.

3.2.3 Results

Figure 3.4 displays the number of epochs (in fact, the logarithm of the number of epochs, given the orders of magnitude involved) required for a standard neural network to learn categories with different frequencies using the minimum square error function.

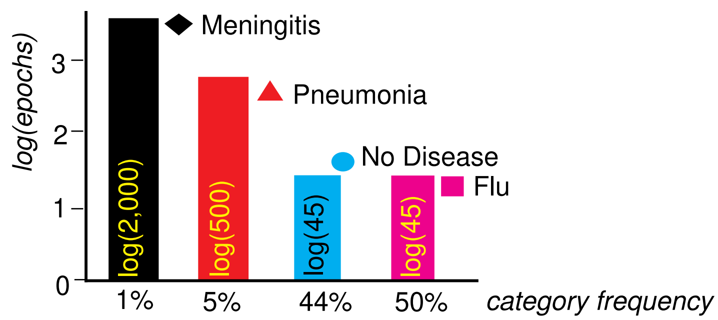
Figure 3.4. Example I: Number of epochs and category frequency.



Frequent categories are learned after a relatively few number of epochs. Infrequent categories take longer to be learned. A 2-4-4 network using the standard error function was used in this example.

Since outputs in this example are binary, the cross-entropy function[4] is the appropriate error function to use in this example [Rumelhart, 1995]. The objective is to maximize the likelihood of a diagnosis given a certain pattern (or approximate the posterior probability of a disease given a pattern). The use of the cross-entropy error function considerably reduces the number of epochs that are necessary to learn each category, as we can see in Figure 3.5, but the relation between the learning times for each category is unchanged: rare categories take more epochs to be learned, and the relation between category frequency and the number of learning cycles is not linear. As we can see in this example, the standard neural network required an overwhelming number of training cycles to detect low-frequency categories.

Figure 3.5. Example I: Number of epochs and category frequency: Using the cross-entropy error function.

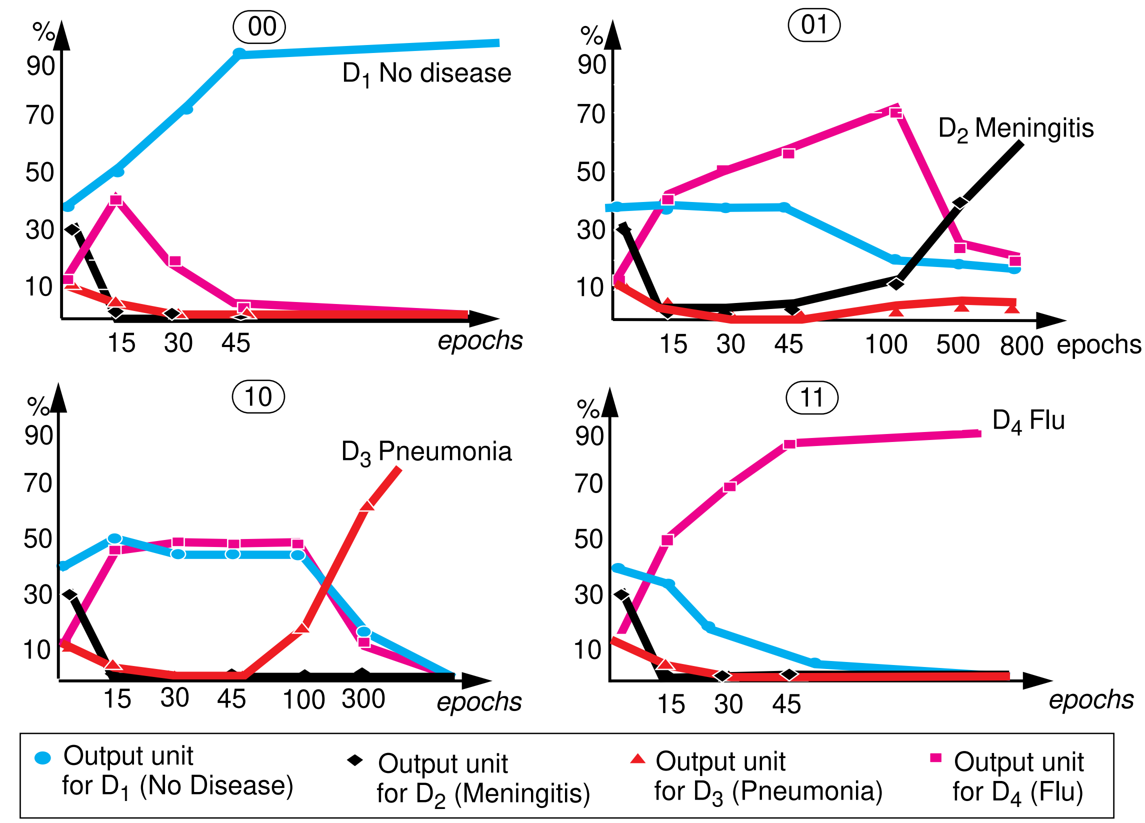


Frequent categories are learned after a relatively few number of epochs. Infrequent categories take longer to be learned. A 2-5-4 network using the cross-entropy error function was used in this example.

Evaluation of a test set was not necessary in this artificial example because the categories were *defined* as being the decimal representation of the binary numbers. The systems would have exhibited the same classification performance on any test set composed of the same patterns, independent of their distribution. Overfitting was not a concern for exactly the same reason. The evolution of the activation values for the output units, when the cross-entropy error function is used, is depicted in Figure 3.6.

[4] The cross-entropy error function is given by $E = -\sum_i [t_i \log y_i + (1 - t_i) \log (1 - y_i)]$,where t_i is the target, or desired output, and y_i is the actual output produced by the network. Minimizing this function is the same as performing maximum likelihood estimation [Curry, 1990].

Figure 3.6. Example I: Activation values for the output units for different input patterns.



Activations of output units start at random values and are updated at each epoch. After enough training epochs, the output unit that corresponds to a given input has higher activation than any other output unit. Note that the network seems to try to guess the most frequent categories as outputs for all input patterns early (~15 epochs), approximating the prior probabilities of the corresponding categories. After this early phase, the categories D_1 and D_4 are easily learned. The same does not happen for D_2 and D_3, since these categories are rare.

Note that it only takes about 30 epochs for pattern “00” (which has a frequency of 44 percent and corresponds to category D_1) to be learned. Compare this number with that of learning pattern “01” (which has a frequency of 1 percent and corresponds to category D2), which is around 800 epochs. Clearly, categories that are rare take longer to be learned. This difference is even more evident when least square errors is performed, as shown previously in Figure 3.4. Note also that, at early stages of learning (e.g., after five epochs of training), the neural network classifies all patterns as belonging to the most frequent category, D_4. In those early stages, the network seems to be only using the prior probability of each category. Later on, the network seems to be able to somehow learn the posterior probability of the disease given S_1 and S_2.

3.2.4 Current methods of recognizing rare categories

We have seen in this example that rare categories take a vast number of epochs to be learned. The two proposed solutions for this problem involve either the preprocessing of inputs or the modification of the error function. Preprocessing involves the replication of patterns in rare categories up to the point where all categories have equal prior probabilities. In this example, since the categorization is deterministic and the classes do not overlap, replicating the patterns in infrequent categories (or, conversely, deleting the patterns in frequent ones) up to the point where each category represents 25 percent of the inputs would result in faster learning, with no decrease in the number of correct results. If the categories are, however, stochastically determined, the change in prior probabilities may result in spurious results, as we will see in the next example.

The second existing solution, the change in error function to account for utilities of learning certain categories, was not attempted, since the neural network was created to produce posterior probabilities of a category given a pattern (or of a disease given the two symptoms), and *not to* produce the optimal decision boundary based on an information-theoretic perspective. Once the posterior probability is produced, the use of risks and the definition of the optimal decision is straightforward. In medicine, one might argue that this is the mechanism used by physicians to make diagnoses. Going back to the trypanosomiasis example presented in Section 1.1, the physician may recognize Chagas' disease as the most probable diagnosis, but may place it low on her differential list, not because the clinical picture is not clear (that is, the posterior probability of that disease given the symptoms is high), but because failing to promptly diagnose this chronic (and at this stage, irreversible), disease is not as harmful for the patient as failing to diagnose other treatable diseases.

3.3 Example II: Probabilistic Sorting of Binary Numbers

Although the replication method seemed to produce good results in Example I, there is an important limitation in the data set used there. All classifications in that example were deterministic and mutually exclusive (that is, once a pattern was known, its classification did not depend on any random factor and it belonged to just one category). I will demonstrate next that when random factors are involved in classification (that is, once a pattern is known, there is a certain probability—different from 1—that it belongs to a given class), the replication method does not work. In this example categories are still mutually exclusive, but the results apply to similar problems where categories are not mutually exclusive as well.

3.3.1 The data set

I have shown in the previous example that backpropagation neural networks take a long time to recognize categories that are infrequent. In that example, the categories were deterministically defined, and the replication method would work nicely to enhance the speed of learning. In this next example, however, categories are stochastic: the same pattern may appear in different categories, and the relative frequency of each pattern in a certain category will determine how that pattern should be classified. The distribution of the patterns and their categories is shown in Table 3.2. The shaded cells indicate the best diagnosis for each input pattern. For example, a patient with symptoms S_1S_2 should be classified as having disease D_4, a patient with $S_1\neg S_2$ should be classified as having D_3, a patient with $\neg S_1S_2$ should be classified as having D_2, and a patient with $\neg S_1\neg S_2$ should be classified as having D_1, because the posterior probability of these diseases is the highest, given the symptoms just mentioned. The posterior probability of a disease given the symptoms is not, of course, 100 percent. For example, a patient with $\neg S_1\neg S_2$ has a probability of 24/44 (54.6%) of having D_1, 2/44(4.5%) of having D_2, 3/44 (6.8%) of having D_3, and 15/44

(34.1%) of having D_4.

Table 3.2. **Distribution of patterns for Example II.**

Input pattern	D_1 (%)	D_2 (%)	D_3 (%)	D_4 (%)	Total (%)
00 $\neg S_1 \neg S_2$ ●	24	2	3	15	44 (44%)
01 $\neg S_1 S_2$ ◆	0	1	0	0	1 (1%)
10 $S_1 \neg S_2$ ▲	1	0	3	1	5 (5%)
11 $S_1 S_2$ ■	12	6	12	20	50 (50%)
Total (%)	37(37%)	9 (9%)	18 (18%)	36 (36%)	100 (100%)

Input pattern	D_1 (%)	D_2 (%)	D_3 (%)	D_4 (%)	Total (%)
00 $\neg S_1 \neg S_2$ ●	●●●●●● ●●●●●● ●●●●●● ●●●●●●	●●	●●●	●●●●●● ●●●●●● ●●●	44 (44%)
01 $\neg S_1 S_2$ ◆		◆			1 (1%)
10 $S_1 \neg S_2$ ▲	▲		▲▲▲	▲	5 (5%)
11 $S_1 S_2$ ■	■■■■■■ ■■■■■■	■■■■■■	■■■■■■ ■■■■■■	■■■■■■ ■■■■■■ ■■■■■■ ■■	50 (50%)
Total (%)	37(37%)	9 (9%)	18 (18%)	36 (36%)	100 (100%)

Shaded cells indicate the best diagnosis for a given pattern (e.g., D_1 is the best diagnosis if "00" is presented, and D_2 is the best diagnosis if "01" is presented. The same information is presented graphically in the lower table. The classification of a pattern (diagnosis) should not change for this data set.

Figure 3.7 shows how patterns are distributed in the four output diagnostic categories.

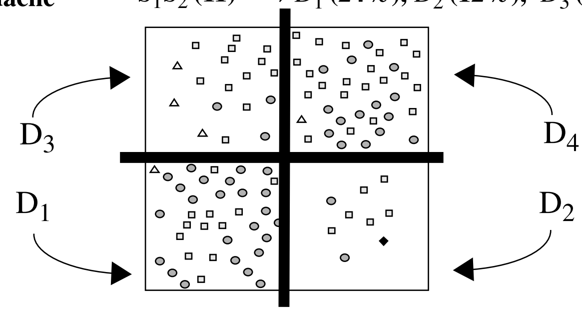
Figure 3.7. Example II: Distribution of patterns and output categories.

¬cough, ¬headache ¬S1¬S2 **(00)** → D1 **(55%)**, D2 **(4%)**, D3 **(7%)**, D4 **(34%)**

¬cough, headache $\neg S_1 S_2$ **(01)** → D_1 **(0%)**, D_2 **(100%)**, D_3 **(0%)**, D_4 **(0%)**

cough, ¬headache $S_1 \neg S_2$ **(10)** → D_1 **(20%)**, D_2 **(0%)**, D_3 **(60%)**, D_4 **(20%)**

cough, headache $S_1 S_2$ **(11)** → D_1 **(24%)**, D_2 **(12%)**, D_3 **(24%)**, D_4 **(40%)**

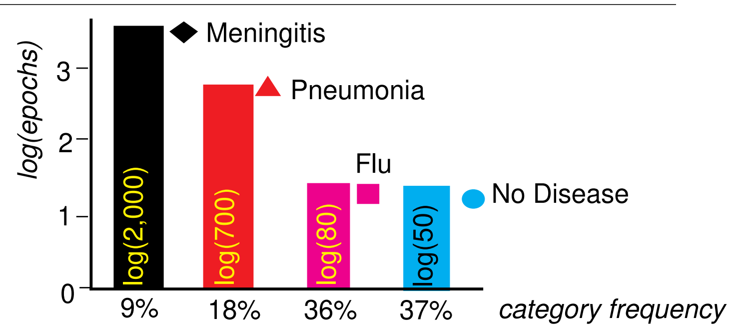


In this artificial data set, classification of diseases is stochastic: each input pattern has a probability of belonging to a given disease category. For example, if pattern "00" is present, there is a probability of 55% that D_1 is present, 4% that D_2 is present, 7% that D_3 is present, and 34% that D_4 is present.

3.3.2 Results

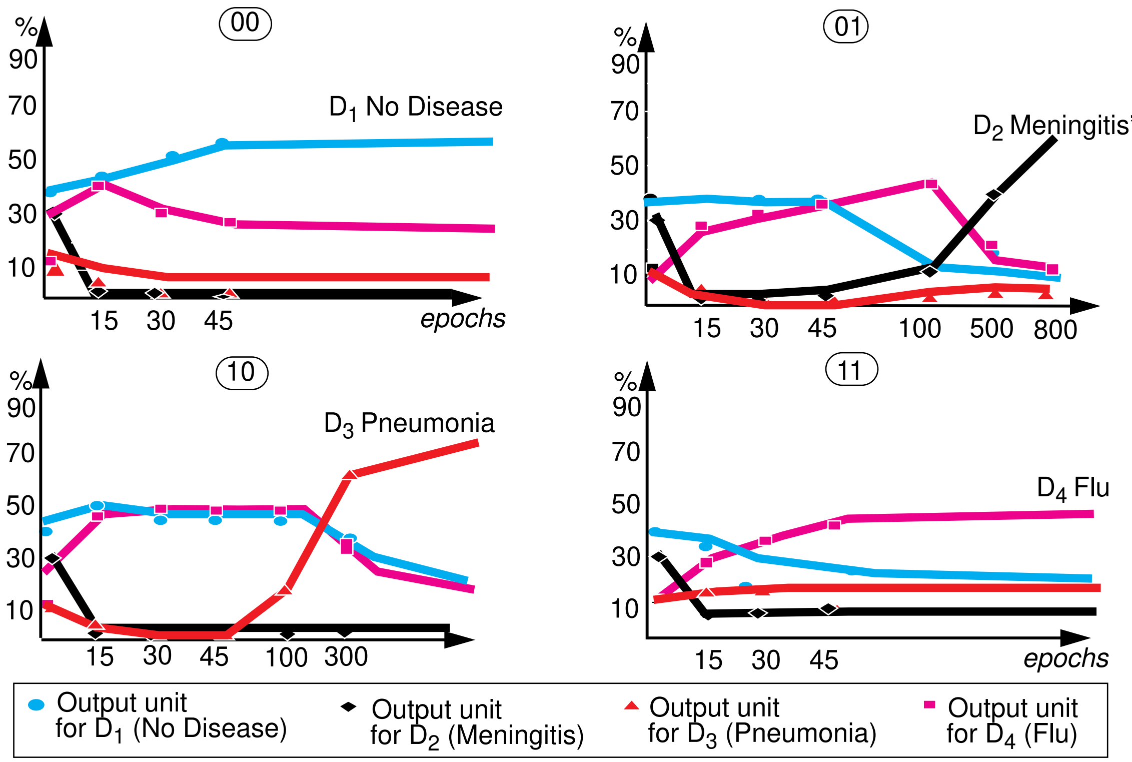
The same network used in Example I, with the cross-entropy error function, was used to classify the patterns. The number of epochs that the network needed to classify correctly all cases is shown in Figure 3.8. The evolution of the activation values for the output units is depicted in Figure 3.9.

Figure 3.8. Example II: Number of epochs and category frequency using the cross-entropy error.



Infrequent categories take longer to be learned. A 2-5-4 network using the cross-entropy error function was used in this example.

Figure 3.9. Example II. Activations for the output units for different input patterns.



Note again that the network tries to guess the most frequent categories as outputs for all input patterns early (~15 epochs), approximating the prior probabilities of the corresponding categories. Infrequent categories (*Meningitis* and *Pneumonia*) are learned later than frequent ones (*No disease* and *Flu*).

The graphics also show the order of a possible differential diagnosis list. For example, if pattern “10” is presented, D_3 would be the first in the list, D_1 and D_4 would be tied in second place, and D_2 would be the most unlikely diagnosis. It is again evident that categories that are less frequent are more difficult for the neural network to recognize.

3.3.3 Replicating patterns in infrequent categories

Now consider the option of replicating some patterns in infrequent categories (so that each output category is equally represented) in order to enhance the speed by which all categories are learned. Table 3.3 shows the distribution after replication of the patterns in the most frequent categories. Figure 3.10 shows graphically how the patterns would be distributed in each category. Patterns are replicated in rare categories, but their proportions inside that category remain practically unchanged after this process. For example, patterns for category D_2 are replicated so that the initial proportion of input patterns — 2/9 (22%) for "00," 1/9 (11%) for "01," 0/9 (0%) for "10," and 6/9 (67%) for "11" are replicated to 8/37 (22%), 4/37 (11%), 0/37 (0%), and 25/37 (67%), respectively.

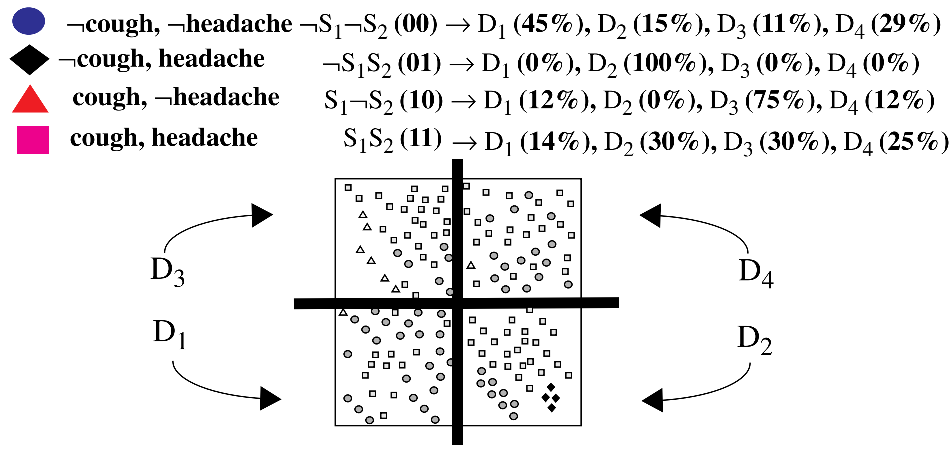
Table 3.3. **Distribution of patterns for Example II, after replication.**

	D_1 (%)	D_2 (%)	D_3 (%)	D_4 (%)	Total (%)
00 $\neg S_1 \neg S_2$ ●	24	8	6	15	53(36%)
01 $\neg S_1 S_2$ ◆	0	4	0	0	4 (3%)
10 $S_1 \neg S_2$ ▲	1	0	6	1	8 (5%)
11 $S_1 S_2$ ■	12	25	25	21	83 (56%)
Total (%)	37 (25%)	37 (25%)	37 (25%)	37 (25%)	148 (100%)

Input pattern	D_1 (%)	D_2 (%)	D_3 (%)	D_4 (%)	Total (%)
00 $\neg S_1 \neg S_2$ ●	●●●●●● ●●●●●● ●●●●●● ●●●●●●	●●●●●● ●●	●●●●●●	●●●●●● ●●●●●● ●●●	44 (44%)
01 $\neg S_1 S_2$ ◆		◆◆◆◆			1 (1%)
10 $S_1 \neg S_2$ ▲	▲		▲▲▲▲▲ ▲	▲	5 (5%)
11 $S_1 S_2$ ■	■■■■■■ ■■■■■■	■■■■■■ ■■■■■■ ■■■■■■ ■■■■■■ ■	■■■■■■ ■■■■■■ ■■■■■■ ■■■■■■ ■	■■■■■■ ■■■■■■ ■■■■■■ ■■■	50 (50%)
Total (%)	37(37%)	9 (9%)	18 (18%)	36 (36%)	100 (100%)

Patterns in infrequent categories were replicated, so that all categories became equally represented (25% of the patterns of each category) in this example. Replication makes learning of a rare category easier. Shaded cells correspond to the diagnosis that should be made for each pattern. The same information is presented graphically in the lower table. Compare these shaded cells with the ones presented in Table 3.2. Note that the best diagnosis for "11" is now either D_2 or D_3, and **not** D_4, *as it should be*. Replication may result in spurious classification when the categories are not deterministically defined, that is, when the same pattern may belong to more than one category.

Figure 3.10. Example II. New distribution of patterns after replication.

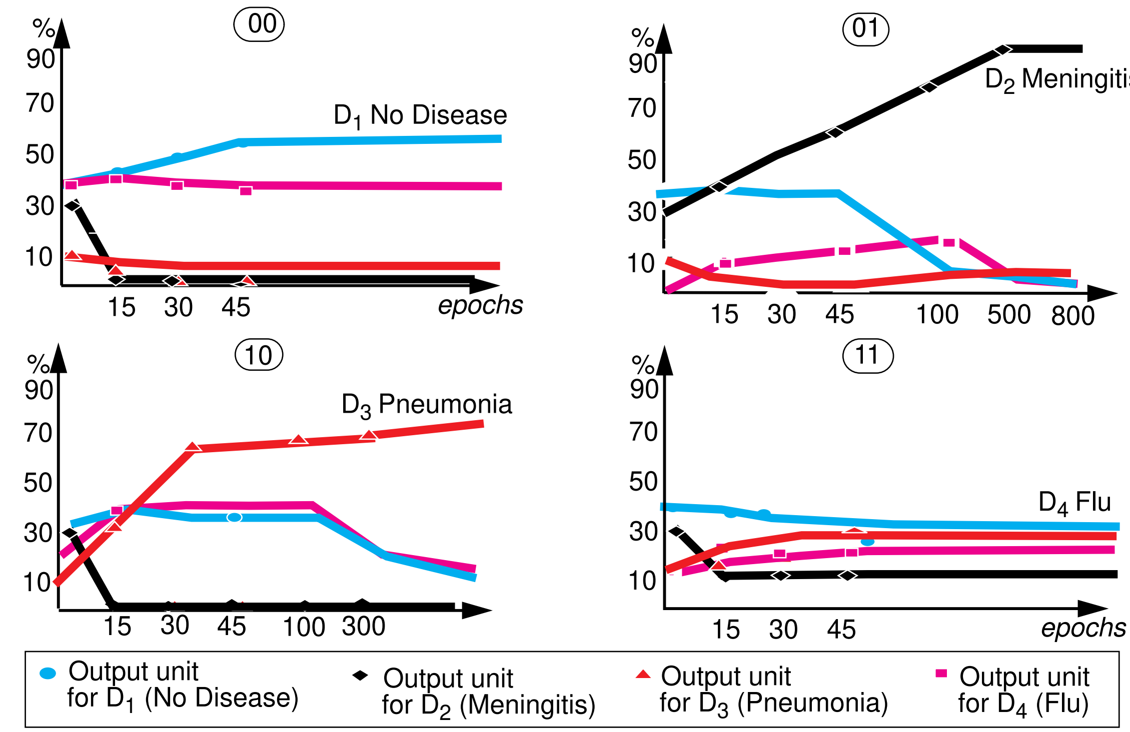


After replication, all quadrants have the same number of patterns. Patterns are replicated in each quadrant so that initial proportions inside a category are approximately unaltered.

Note that changing the prior probabilities of each category by replicating patterns in that category will cause a change in posterior probabilities. Since the network approximates the posterior probabilities given an input pattern, it will produce wrong results. For example, Figure 3.11 shows the activation of the output units during learning.

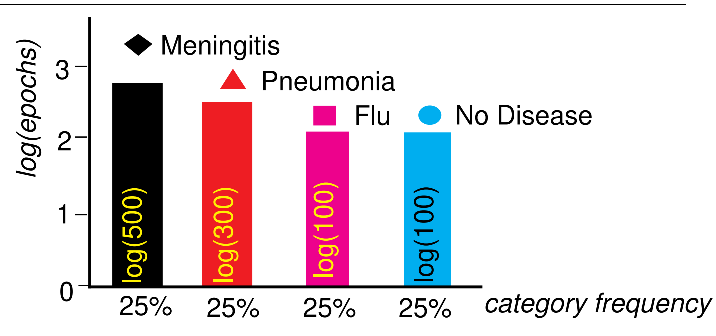
The network does not classify correctly cases with input pattern "11": it tries to guess that the output for that pattern should be either D_2 or D_3, whereas in fact it should be D_4 (the correct diagnosis before replication). Therefore, learning may be faster with replication, but accuracy is sacrificed. The number of false positives for categories in which input patterns were replicated rises. The number of epochs that the network needed to classify correctly all cases is shown in Figure 3.12.

Figure 3.11. Example II: Activation values for the output units for different input patterns after replication.



After replication, learning categories is apparently easier. After 45 epochs, all categories are learned. However, since posterior probabilities of each category given an input pattern have changed, the categorization is not correct in all cases. For example, whenever pattern "11" is presented, the most probable diagnosis now is either D_2 or D_3, and not D_4, which is not correct. The number of false positives for D_2 and D_3 is now higher.

Figure 3.12. Example II: Using the cross-entropy error function after replication.



After replication, the difference in the number of epochs that are necessary to learn each category is smaller than before replication. A 2-5-4 network using the cross-entropy error function was used in this example.

3.3.4 Removing patterns in frequent categories

Removal of patterns from frequent categories so that all categories become equally represented has the same effect as replicating patterns. Furthermore, information contained in the removed cases is lost. A minimum of one case had to be left for each input pattern that corresponded to a given diagnosis in the initial sample. Table 3.4 shows the distribution of cases after removal. Shaded areas correspond to the best diagnosis given this distribution. Note that the diagnosis for pattern “11” is again incorrect. Pattern “10” will also be incorrectly classified.

Table 3.4. **Distribution of patterns for Example II, after removal.**

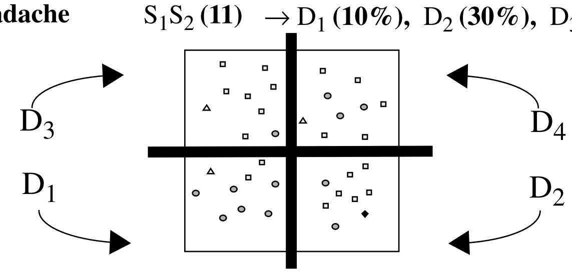
	D_1 (%)	D_2 (%)	D_3 (%)	D_4 (%)	Total (%)
00 $\neg S_1 \neg S_2$ ●	6	2	1	3	12 (33%)
01 $\neg S_1 S_2$ ◆	0	1	0	0	1 (3%)
10 $S_1 \neg S_2$ ▲	1	0	1	1	3 (8%)
11 $S_1 S_2$ ■	2	6	7	5	20 (56%)
Total (%)	9(25%)	9 (25%)	9(25%)	9(25%)	36(100%)

In this example, patterns in frequent categories were removed so that all categories became equally represented (25% of the patterns in each category).

Figure 3.13 shows graphically how the patterns would be distributed in each category

Figure 3.13. Example II. New distribution of patterns after removal.

● **¬cough, ¬headache** $\neg S_1 \neg S_2$ **(00)** $\rightarrow$ D_1 **(50%)**, D_2 **(16%)**, D_3 **(9%)**, D_4 **(25%)**
◆ **¬cough, headache** $\neg S_1 S_2$ **(01)** $\rightarrow$ D_1 **(0%)**, D_2 **(100%)**, D_3 **(0%)**, D_4 **(0%)**
▲ **cough, ¬headache** $S_1 \neg S_2$ **(10)** $\rightarrow$ D_1 **(33%)**, D_2 **(0%)**, D_3 **(33%)**, D_4 **(33%)**
■ **cough, headache** $S_1 S_2$ **(11)** $\rightarrow$ D_1 **(10%)**, D_2 **(30%)**, D_3 **(35%)**, D_4 **(25%)**



After removal, all quadrants have the same number of patterns. Patterns are removed from each quadrant so that initial proportions are approximately unaltered. Fractions of a pattern were rounded to 1.

3.4 Summary

I have shown in an artificial example that learning rare categories in a backpropagation neural network is hard. Currently available solutions are not adequate. By artificially increasing the prior probability of a category, in the case of rare-category replication (or removal of frequent categories), the rise in sensitivity for a given category is coupled with a decrease in specificity. If utilities are used to make learning of one category faster than the learning of others, in the case of error function modification, retraining of the network is necessary every time the utilities change.

In screening large databases, where the desired category is rare, it is desirable that the rate of false positives for that category not be too high. Therefore, there is a need to improve sensitivity to rare categories, without a corresponding decrease in specificity. Hierarchical neural networks are good candidates for achieving this goal, as we will see in Chapter 4.

CHAPTER 4

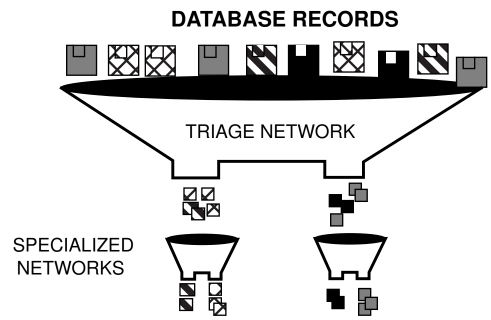
Hierarchical Neural Networks for Diagnosis

This chapter presents a new solution to the problem of recognizing infrequent categories, discussed in Chapter 3. A hierarchical system of triage and specialized networks is applied to the same problems described in Chapter 3, demonstrating an improvement in learning time. Section 4.1 describes hierarchical systems of neural networks (HNNs), and how the use of intermediate abstractions to divide and conquer the problem provides a means to enhance sensitivity to rare categories, without a corresponding decrease in specificity. Section 4.2 describes hierarchical structures of neural networks that have been developed by other authors, emphasizing major similarities and differences with this work. Section 4.3 uses the example of the deterministic sorting of binary numbers to illustrate the advantages of hierarchical neural networks. Section 4.4 is analogous to Section 3.3 in Chapter 3, where a similar data set is used for learning probabilistic classification. Section 4.5 uses a real-world complex medical diagnostic problem to illustrate the use of HNNs.

4.1 Using Divide-and-Conquer in Neural Networks

The HNN is an architecture of neural networks in which the problem is divided and solved in more than one step. Figure 4.1 shows how a hierarchical system of neural networks should operate: the first classifier, or triage network, divides the data set into smaller subsets, which will then constitute the inputs for the specialized networks.

Figure 4.1. Hierarchical neural network.



Electronic data from medical records are entered into a triage network. This network filters records that should be further processed by specialized networks.

The application of multilayered neural networks in more than one step allows the prior probability of a given category to increase at each step, provided that the predictive power of the network at the previous level is greater than that of simply guessing the most frequent category (i.e., the area under the ROC curve is greater than 0.5). For example, suppose a researcher needs to discriminate four categories in a given data set. Among the categories, there exists one that occurs only 1 percent of the time. The other categories have prior probabilities of 5, 44, and 50 percent (see the example in Chapter 3). By applying a classifier that can reliably discriminate a set of two categories from the other ones, and applying another classifier to the results of this preclassification, the total number of categories in the second step is decreased, and consequently the frequency of a given category is increased. This increase in frequency allows a hierarchical neural network classifier to learn to discriminate categories more quickly, as I will demonstrate.

The hierarchical model assumes that the first classifier (triage network) is able to discriminate a superset of some categories, which includes the desired one, from the other categories. Since in any of these reliably constructed supersets the prior probability of a category in the set is higher than that of the initial sample, this process will yield higher posterior probabilities for the desired category than one in which the classifier attempts to make all distinctions in a single step.

Using Bayes's rule, where X is a vector of attributes and C_i is a category, we have

$$P(C_1|X) = \frac{P(X|C_1)P(C_1)}{\Sigma P(X|C_i)P(C_i)}$$

In the two-category case, the equation becomes

$$P(C_1|X) = \frac{P(X|C_1)P(C_1)}{P(X|C_1)P(C_1) + P(X|\neg C_1)P(\neg C_1)}$$

Assuming that $k_1 = P(X|C_1)$ and $k_2 = P(X|\neg C_1)$ are constants and that $P(\neg C_1) = 1 - P(C_1)$, we can see that whenever $P(C_1)$ is increased, the posterior probability $P(C_1|X)$ is also increased, as shown in

$$P(C_1|X) = \frac{k_1 P(C_1)}{(k_1 - k_2)P(C_1) + k_2}$$

Therefore, if the prior probability of a category is augmented in the training set and the sensitivity and specificity of the network remain unchanged, the posterior probability of the class is increased. In other words, if the triage and the specialized networks of the hierarchical system each have the same number of weights as that of the generic system (and consequently the same potential for achieving the same sensitivity and specificity after training), they can perform better than the nonhierarchical system can.

This process confirms the intuition that if the prevalence of a category is increased, while everything else remains unchanged, the posterior probability of that category, given the same set of attributes, is increased. Therefore, if a triage network is applied and is able to reliably discriminate a subset that contains the desired category, an increase in the prior probability of that category will occur, also causing an increase in the posterior probability of that category in the corresponding specialized network. The question remains whether triage and specialized networks with a smaller number of weights than that of the corresponding generic network (a network that classifies patterns in just one step) can also perform better than the nonhierarchical system. If the *total* number of free parameters (weights) in both systems is the same, the triage and specialized networks will certainly have *fewer weights* than the generic network. The experiments described in this chapter were designed to answer this question.

4.2 Hierarchical Architectures

Several authors have dealt with the decomposition of complex problems inside and outside the field of neural networks. Their reasons for developing hierarchical models of neural networks were sometimes different from the ones presented in this work.

Before commenting on existing hierarchical models and describing the details of the experiments used throughout this dissertation, it will be helpful to define some terms that will be used frequently.

Model: A set of coefficients resulting from parameter estimation using statistical models (logistic regression or Cox proportional hazards) or a set of weights learned by a neural network.

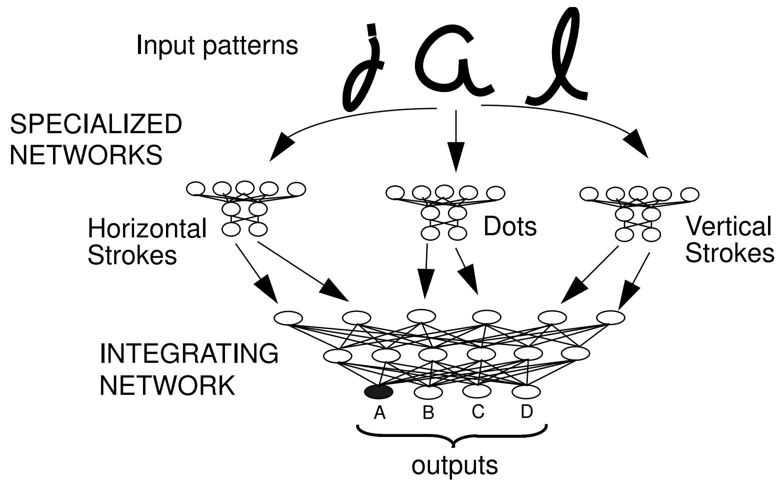
Method: A way of constructing models. A standard method makes only final classifications of specific diagnoses. A hierarchical method makes intermediate classification of cases in classes, and then final classification of cases in specific diagnoses.

Hierarchical systems can be constructed either in a bottom–up or a top–down manner, as explained next.

4.2.1 Bottom-up hierarchical architectures

In bottom-up hierarchical designs, several specialized networks are used to classify all instances, and the results of these specialized networks are aggregated by a top-level network. Usually, the specialized networks work only on certain features. For example, in a handwritten-character–recognition task, one specialized network may be used to identify horizontal strokes, while another may be used to identify vertical ones. A top-level network integrates the results of these specialized networks and provides the final solution. In this type of design, all instances are used in all networks, as illustrated in Figure 4.2.

Figure 4.2. Bottom-up hierarchical architectures.



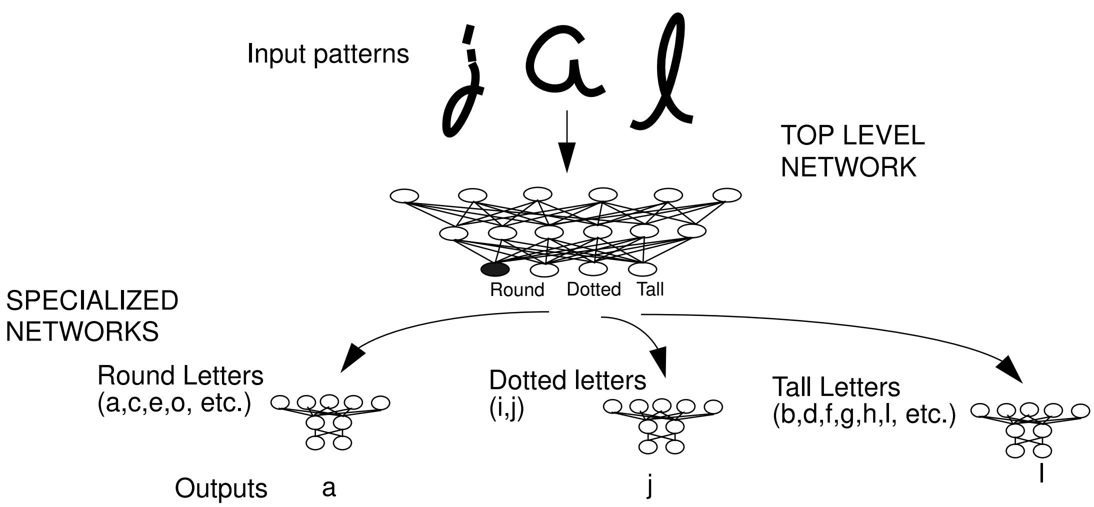
Specialized networks classify inputs according to a specific feature. The results of the specialized networks are used by an Integrating network that provides the final output.

There are several examples of bottom-up hierarchical architectures. Fukushima [1988] developed the Neocognitron for eliminating the problem of space variations in the visual recognition of handwritten digits. The author was not specifically concerned with the frequencies of the categories involved. He has also suggested that there are similarities between his architecture and the human visual cortex. Ballard [1990] also developed a system of hierarchical neural networks for applications in machine vision, and he was particularly concerned with the problem that the backpropagation algorithm might not scale up to complex networks. Hrycej [1992] discussed modularization in neural networks. In his system, preprocessing of inputs was done in an unsupervised manner by a neural network, and the results of this factoring process were then imputed in the following networks. Hripcsak [1990] developed a connectionist model for decision support in medicine based on several backpropagation modules to incorporate real-valued and uncertain data. Jordan [1991] proposed a system in which many networks of experts would receive the inputs and compete for providing the best solution. A gating network would then decide among the experts' solutions. The system used in this work is different: Even though specialized networks refine the partial solutions proposed by the triage network, the decision on which network to use is done first, so not all experts need to be overburdened with all data. It follows a top-down hierarchical architecture.

4.2.2 Top-down hierarchical architectures

Top-down hierarchical systems are different from bottom-up hierarchical systems not only in the way they are created, but also in the final neural network design. In this type of hierarchy, a top-level network divides the inputs to be classified in specialized networks. Figure 4.3 shows the design of this type of systems.

Figure 4.3. Top-down hierarchical architectures.



In top-down hierarchies, all inputs are preclassified by a top-level network to be used in specialized networks that provide the final classification.

Matsuoka et al. [1989] used a top-down hierarchical architecture to perform syllable recognition, and showed certain advantages over a nonhierarchical backpropagation neural network constructed to perform the same task. The authors were not concerned with the problem of rare-category recognition and did not try to generalize their results to other applications. Furthermore, the authors seem to have used the same data set to train and to test their results. The same criticism applies to the work of Cho and Kim [1990] for printed character recognition. Other authors have employed similar architectures to biological data, but have not explored the gain in accuracy over standard models [Boddy, 1994].

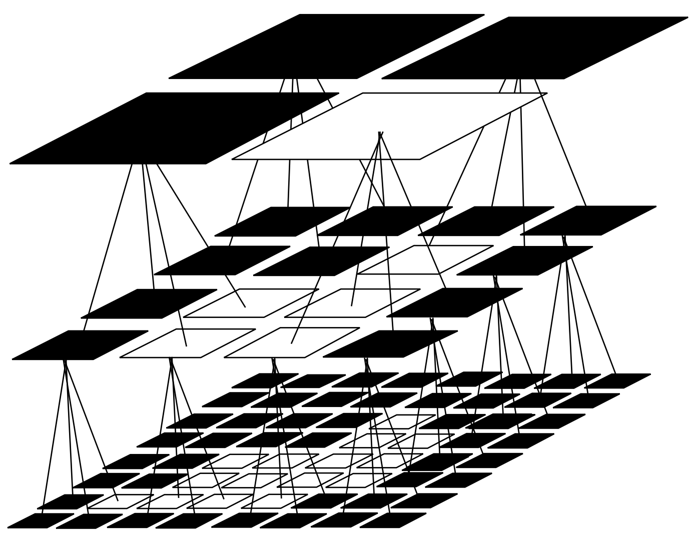
Curry and Rumelhart's work on the Mass Spectrometry Network (MSNet) is closely related to the work presented here. In that system, categories of chemical compounds were determined in a top-level network. The probability of belonging to a given group, allied to the original input attribute vector, was then used by specialized networks to refine the solution and get a final diagnosis. The authors were concerned with the fact that low-frequency categories would cause the performance of the network to decay, and they addressed that concern by using a different strategy: they trained the network to recognize low-frequency categories by assigning a utility to these categories. A factor inversely proportional to the frequency was multiplied by the error originating from each category, so that the network was trained as if all categories were equally represented. This procedure was done by modifying the learning algorithm and processing the final output to reflect the consequent changes in posterior probabilities.

The system of HNNs that I propose, however, tries to disambiguate the process of diagnosing the categories from the process of using utilities while training to make an optimal decision based on a decision-theoretic approach. In this system, the diagnosis is based on the similarities among the categories, and not on their relative utility. Once the diagnostic process is proven to be reliable and is based mainly on the features presented by the inputs, the use of utilities and the decision on which category to choose should be straightforward.

4.2.3 Theory of hierarchical modular systems

Large self-organizing natural systems, such as monetary, computer vision, and natural-language understanding systems, can be modeled using hierarchical modular systems . In computer vision, pyramidal architectures (usually binary or quad pyramids, shown in Figure 4.4) have been used to segment images for machine recognition. Variable-resolution grids compose multiresolution systems with layers of increasing detail [Cantoni, 1994]. This computer architecture tries to mimic human vision, which is composed of a *preattentive phase*, in which detection of regions of interest is performed, and an *attentive phase*, in which extensive analysis of a subset of the visual field is performed [Freeman, 1988]. In pyramidal architectures for image segmentation [Bischof, 1995], the initial image is divided into quadrants. A significant change in color from one quadrant to another signals that segmentation should take place somewhere between those quadrants. Finer grids (subquadrants) are therefore created in each of these quadrants, and the process is repeated. Adjacent quadrants with no color difference are ignored for purposes of segmentation, so that the use of subquadrants is not widespread, causing significant resource savings.

Figure 4.4. Pyramidal architectures for computer vision.



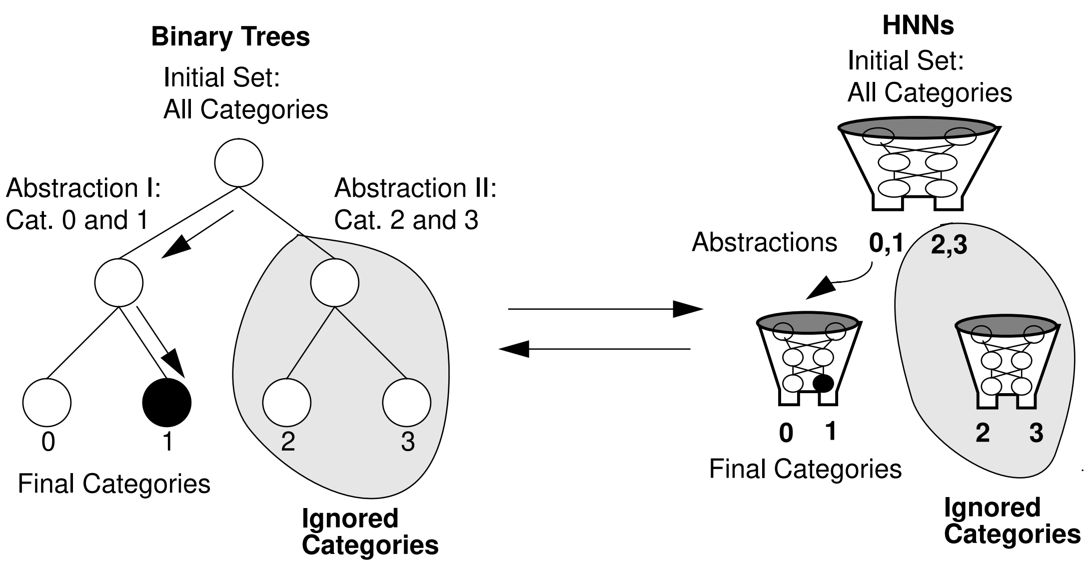
At the top level, segmentation is coarse: the system can detect only gross changes in color from one quadrant to another. Adjacent quarters that have no color change will probably not contain borders and will therefore be ignored by the segmentation algorithm. Adjacent quadrants that have a color difference will be subdivided into subquadrants, and the algorithm will be recursively applied until the borders are well defined. At the bottom of the pyramid, segmentation is fine.

HMS can be adapted for use with neural networks. In hierarchical neural networks, intermediate abstractions (analogous to the quadrants used in computer vision) need to be defined in order to make learning faster. In medical applications, however, there is usually an additional goal: the intermediate abstractions need to make sense from a medical perspective, so that if learning had to be interrupted at any level, the results of the classification up to that level would be medically useful, and the work would not have been a loss.

4.2.4 Divide-and-conquer methods

Divide-and-conquer methods are not new in computer science. Simple search trees are perhaps the best-known example, where the complexity of simple operations takes *O(log n)*, with *n* being the number of candidate solutions, as opposed to *O(n)*, as occurs with linear lists [Ullman, 1993]. We can make an intuitive assessment of the value of hierarchical systems when the objective is to recognize only one category of rare outputs in a set of three or more categories if we think of the hierarchical architecture as a binary tree, as shown in Figure 4.5.

Figure 4.5. Hierarchical decomposition: binary trees and HNNs.

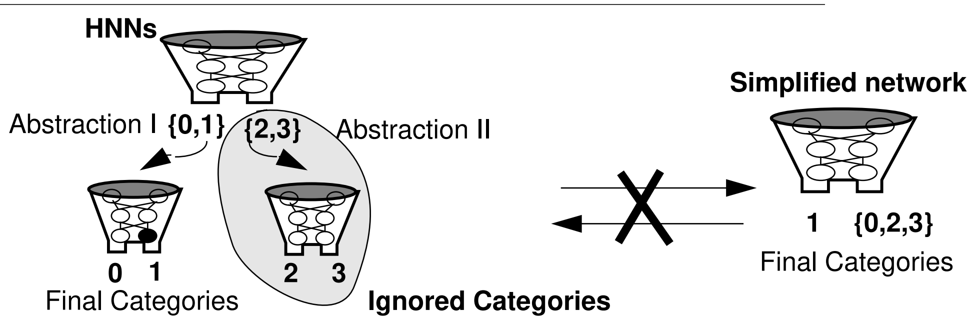


When the objective is to learn one category (in this example, category 1), certain HNNs can be viewed as binary trees: the solution space is recursively divided in halves, and categories that do not belong to the path are ignored.

In this case, we are comparing the task of learning all categories in the nonhierarchical system to that of learning only one category in the hierarchical case (since we successively discard all cases not belonging to that category or any supersets containing that category). This would be the equivalent of doing a single search in a binary tree in which all leaf nodes were the categories (*O(log n*), where *n* is the number of categories), as opposed to traversing a linked list of categories, as in the nonhierarchical case (*O(n)*). As in binary trees, the objective in a HNN is to make the tree as balanced as possible, so that the paths that lead to leaf nodes are short [Cormen, 1990].

One could argue that if the only objective is to learn a given category, a simplified nonhierarchical neural network could be built that would only distinguish between the desired category and “others,” as shown in Figure 4.6. This simplified network would then be solving the identical problem as the HNN, and therefore should take a similar amount of time. This is sometimes not verified in practice (the simplified network seems to take either the same number as the HNN or more cycles than the HNN for the problems presented in Chapter 3). The reason may be that the frequency of the desired category in the simplified network is lower than the frequency in any network of the hierarchical system. Furthermore, the results of the simplified network may not always be useful when the goal becomes to learn another category, as opposed to what happens in HNNs, where the intermediate results can always be reused.

Figure 4.6. Decomposition: HNNs and simplified nonhierarchical neural networks.



It could be argued that a simplified nonhierarchical neural network could learn to distinguish category 1 from the other categories faster than an HNN can. This fact was not observed in the example presented in Chapter 3, because the frequency of category 1 was very low in the nonhierarchical example.

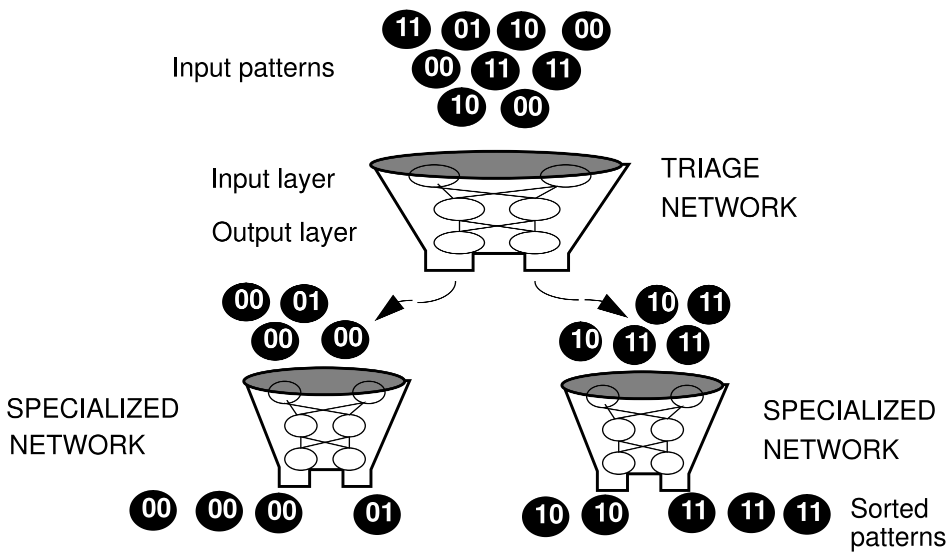
4.2.5 How to define intermediate abstractions

There are basically three approaches to grouping categories to create intermediate classifications (or *intermediate abstractions*) for use in the triage networks: (1) a conceptual approach, in which categories are classified according to predefinitions imposed by the domain (e.g., in medicine, the classification of patients into “normal,” “hypothyroid,” and “hyperthyroid” is a conceptual one, and will be used in Chapter 5); (2) a utility-based approach, in which categories are classified such that the cost of making a misclassification *within* a group is much lower than that of misclassifying cases *between* groups; and (3) a pattern-recognition approach, in which categories are grouped in superclasses (intermediate abstractions) according to the similarity in their features (e.g., patterns “011” and “111” are similar and should be grouped together, whereas “100” and “000” should belong to a different group). The decision as to which approach is the best depends on the application. All approaches have their advantages and drawbacks. Approach (1) can be easily justified only in a domain where the concepts are well defined, but is easily understood by the user; approach (2) maximizes the utility of grouping, but it is based on how the information on the classification can be best used, and therefore requires different groupings every time utilities change; approach (3) seems to work well, as discussed in the next section, but requires a clustering algorithm to define intermediate abstractions that may not make sense to the user.

4.3 Example I: Deterministic Sorting of Binary Numbers

I tested the hypothesis that the HNN could discriminate low-frequency categories earlier (i.e., requiring fewer training cycles) than a standard neural network could, provided that the systems had the same number of weights, using the same data set described in Section 3.2. Figure 4.7 shows how the hierarchical system of neural networks works. The standard feed-forward neural network that tries to classify the categories in just one step was used for comparison. Classification in the HNN was done in a supervised manner in each step. The neural networks of the first level (triage networks) discriminate categories 0 and 1 (patterns “00” and “01”) from categories 2 and 3 (patterns “10” and “11”). The two networks of the second level (specialized networks) discriminate patterns between the categories 0 and 1 and categories 2 and 3, respectively.[1] Note that the *total* number of weights in the HNN is approximately the same as that of the standard neural network (i.e., the total number of parameters that needed to be estimated in each of the systems is controlled to be approximately the same).

Figure 4.7. Hierarchical sorting of binary numbers.



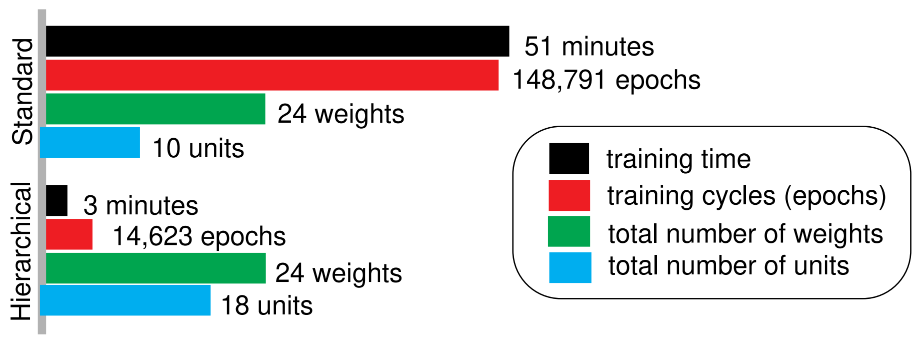
Instead of sorting all numbers at once, this hierarchical system of neural networks first decides which inputs are either “00” or “01,” and sends them to a specialized network that sorts “00” and “01.” Other inputs are sent to another specialized network that sorts “10” and “11.”

Figure 4.8 displays the number of parameters (weights) to be estimated, the number of training cycles (epochs), and the average time that each system took to converge to a perfect solution. A perfect solution was defined to be achieved when the activation of the correct output unit was at least twice that of the other output units. No noise was added to the data. Training was done by epochs. I performed 10 simulations for each system, starting

1. A category name (output, such as 0) corresponds to the decimal value of a binary number, which is represented as the input pattern (e.g., “00”).

with different initial weights. All networks were trained with a fixed learning rate of 0.01 and no momentum term. The overall time spent on making the perfect classification was significantly reduced ($p < 0.01$) with the use of HNN. I did not run specialized networks in parallel, even though by doing so the amount of time could be reduced even more. It must also be taken into account that one epoch in the nonhierarchical network takes longer than one epoch in any of the networks in the hierarchical system, given the smaller number of weights in each of the networks of the latter, and the smaller number of patterns in the specialized networks.

Figure 4.8. Comparison of systems: Standard error function.

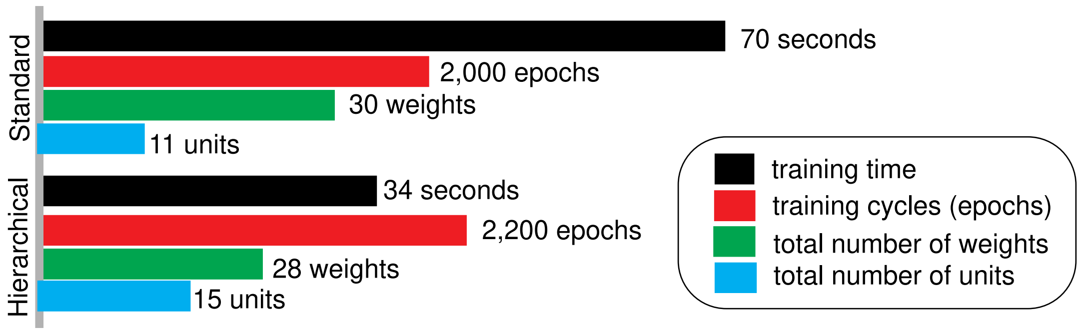


Hierarchical systems learn infrequent category 1 (pattern "01") faster than do standard nonhierarchical systems.

The same experiment was repeated using the cross-entropy error function (defined in Section 3.2) and triage network with 2 input, 3 hidden, and 2 output nodes (2-3-2), and a specialized network with 2 input, 4 hidden, and 3 output nodes (2-4-3). The total number of weights in the HNN was 28 (12 for the triage and 8 for the two specialized networks), compared to 30 for the (2-5-4) nonhierarchical network presented in Section 3.3.

Results using the cross-entropy error function were very similar to the ones just presented for the standard error function, as shown in Figure 4.9. As expected, since the output units were binomial, the use of the cross-entropy error function increased speed of learning in both the nonhierarchical and the hierarchical systems, when compared to that of the standard error function [Rumelhart, 1995]. However, the hierarchical system was still able to learn infrequent categories faster than the nonhierarchical system.

Figure 4.9. Comparison of systems: Cross-entropy error function.



Hierarchical systems could learn infrequent categories faster than could standard nonhierarchical systems.

The perceptron's performance on this problem was extremely good, as expected, but it would not be as good in the case of a nonlinearly separable problem, as we will see in Example II. A multilayered neural network that has enough hidden nodes can approximate any function [Hornik, 1989], and its applicability is therefore much broader than that of a perceptron. Furthermore, a hierarchical system of perceptrons also proved to converge faster than a standard perceptron did in this example, as shown in Figure 4.10.

Figure 4.10. Comparison of perceptrons.

A hierarchical system of perceptrons learned faster than a standard nonhierarchical system.

One might still argue that the preselection of subsets that were themselves linearly separable introduced a bias in favor of the hierarchical system. I also ran the same experiments dividing the subsets in a different way, such that categories 0 and 3 (patterns "00" and "11") would be separated from categories 1 and 2 (patterns "01" and "10") in the triage network. This grouping required the triage network to solve a nonlinearly separable problem first, and is by far the worst possible grouping: the Hamming distance[1] between patterns in the same group is twice that of patterns in other groups. Furthermore, the proportions involved required the triage network to detect a subgroup that had a low-frequency value itself (categories 1 and 2, corresponding to patterns “01” and “10,” constitute only six percent of the total number of patterns). Ten systems of networks, starting with different initial random weights, were built. The HNN exhibited a peculiar behavior: four of the ten systems converged to a solution after relatively few epochs (mean: 34,944), but the other six did not converge to a perfect solution even after 4×10^5 epochs. This result indicates that patterns should be grouped by similarity of features, rather than be based purely on category frequencies, for the triage network to work. Merging rare categories that do not share similarities into a group simply to increase their frequency in the training set does not seem to enhance the performance of the triage network.

Another experiment, in which the pattern distribution was changed to the one shown in Table 4.1, also indicated that the difficulties encountered by the triage network were not related to the combined low frequency of the group “01” and “10,” but to the fact that the similarities within the groups were low. None of the 10 triage networks built for this experiment converged to a perfect solution after 4×10^5 epochs. Pattern similarity seems to be the key factor in determining the success of the HNN.

Table 4.1. Another distribution of patterns for Example I.

Pattern	Frequency	Output (Category)
00	1%	0
01	45%	1
10	5%	2
11	49%	3

1. The Hamming distance is the difference in the number of bits of two patterns. For example, the Hamming distance between “00” and “10,” “10” and “11,” or “00” and “01” is 1, whereas the difference between “10” and “01” or “00 and “11” is 2.

4.4 Example II: Probabilistic Sorting of Binary Numbers

In order to check whether the performance of HNNs would be better than that of the hierarchical system when classification is stochastic, the following experiment was designed. I used the same data set provided in Section 3.3 to test the hypothesis that the HNN could learn the classification of D_2 earlier than the nonhierarchical system, without a decrease in specificity for that category. Table 4.2 shows the distribution of patterns for the triage network.

Table 4.2. Triage network: Distribution of patterns for Example II.

Input pattern	D_1 or D_2 (%)	D_3 or D_4 (%)	Total(%)
00 $\neg S_1 \neg S_2$ ●	26	18	44 (44%)
01 $\neg S_1 S_2$ ◆	1	0	1 (1%)
10 $S_1 \neg S_2$ ▲	1	4	5 (5%)
11 $S_1 S_2$ ■	18	32	50 (50%)
Total (%)	46(46%)	54 (54%)	100 (100)

Shaded cells indicate the most frequent category for a given pattern. The first column indicate the input patterns. The second column indicates how many patterns belong to categories D_1 or D_2. The third column indicates how many patterns belong to categories D_3 or D_4.

Table 4.2 shows the distribution of patterns for the specialized network that decides between 00 and 01. Patterns 10 and 11 were sent to the other specialized network.

Note that the frequencies for category D_2 are much higher in the first specialized network (7%) than those in the nonhierarchical system (1%).

Table 4.3. Specialized network: Distribution of patterns for Example II.

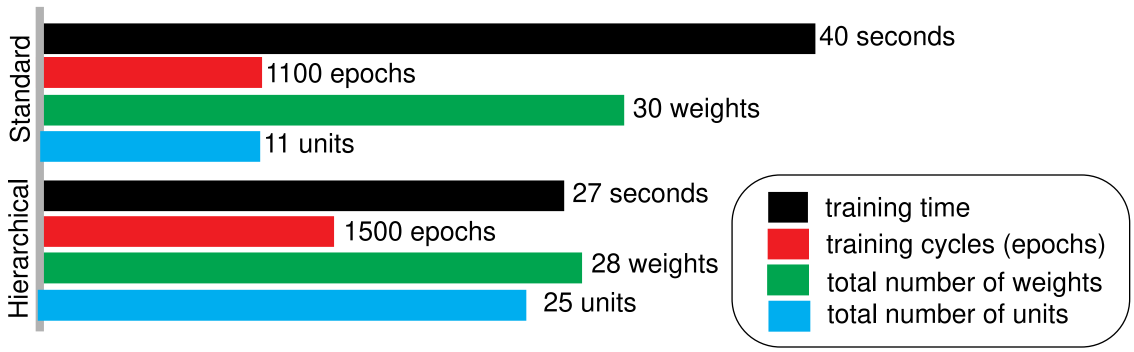
Input pattern	D_1 (%)	D_2 (%)	Others: D_3or D_4 (%)	Total (%)
00 $\neg S_1 \neg S_2$ ●	24	2	18	44 (98%)
01 $\neg S_1 S_2$ ◆	0	1	0	1 (2%)
Total (%)	24(53%)	3 (7%)	18(40%)	45(100%)

Shaded cells indicate the most frequent category for a given pattern.

Figure 4.11 shows the comparison of an HNN and a nonhierarchical neural network for this example. The difference in time was not as remarkable as in the deterministic example, but the less frequent category D_2 was not as rare in Example II as in Example I (9%

and 1%, respectively). HNNs seem to be more advantageous when frequencies of the category one wants to detect are very low.

Figure 4.11. Example II: Comparison of systems using the cross-entropy error function.



Hierarchical systems learn infrequent categories faster than do standard nonhierarchical systems in a stochastic classification task.

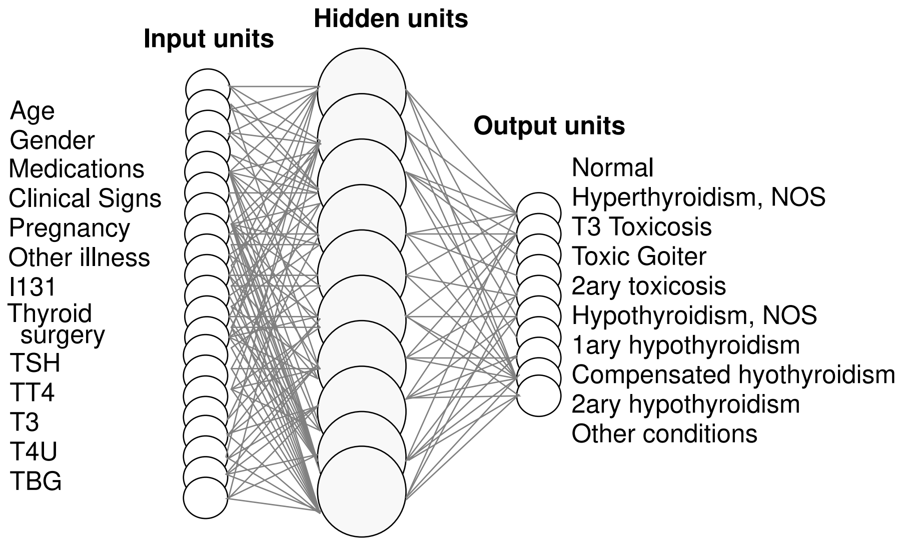
Other decompositions were also used, and in all of them the learning performance in terms of time of the HNN was at least as good as the nonhierarchical system.

4.5 Example III: Diagnosis of Thyroid Diseases

Thyroid diseases result from hyper- or hyposecretion of thyroid hormones and have a high prevalence in the U.S. (2 – 3%) [U.S. Preventive Services Task Force, 1989]. The diagnosis of thyroid conditions is based on interpretation of clinical and laboratory findings. Accurate diagnosis of major classes of thyroid diseases, such as hyperthyroidism and hypothyroidism, has been done accurately by some automated methods. None of these methods, however, has shown results on refined diagnoses such as primary, secondary, or compensated hypothyroidism.

For my experiments, I obtained a set of 9172 patients suspected of having thyroid diseases from the data repository at the University of California at Irvine [Murphy, 1993]. The same data set was used by Quinlan to demonstrate the performance of decision trees in diagnosing hypothyroidism [Quinlan, 1986]. I used a subset of 4,586 patients to train the networks. A standard neural network discriminated 10 different diagnoses. It consisted of 22 inputs, 10 hidden units, and 10 outputs. The standard neural network, or generic network, is shown in Figure 4.12.

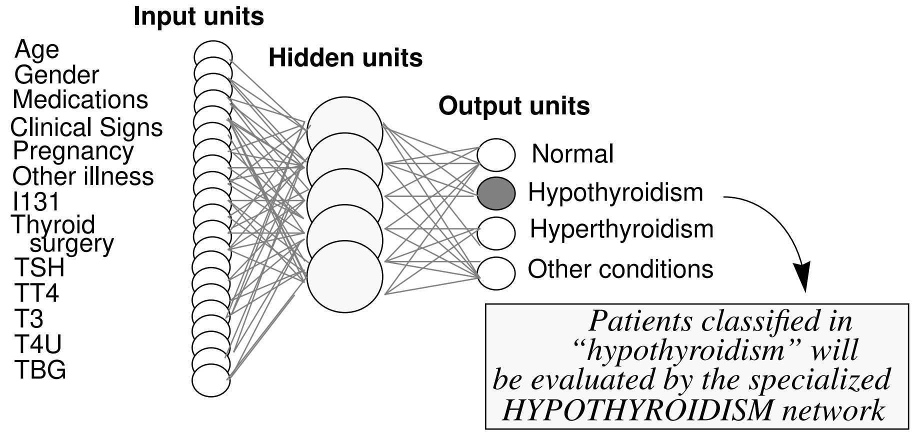
Figure 4.12. A generic neural network for thyroid disease.



This network tries to diagnose 10 different diseases in just one step.

In the HNN, the triage network was dedicated to discriminating patterns of *hypothyroidism, hyperthyroidism, normality,* and *other thyroid conditions*. The rationale for establishing these groupings was based on the assumptions that (a) patients in each group shared similar attribute values and (b), even if not all the specialized networks were able to refine the solution and obtain a final diagnosis, the partial diagnoses provided by the triage network could be clinically useful. Figure 4.13 shows the triage network.

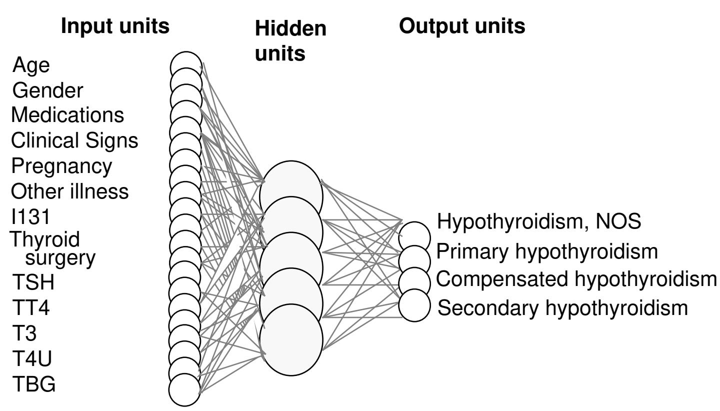
Figure 4.13. A triage neural network for thyroid disease.



This network separates the patterns corresponding to "Normal," "Hypothyroidism," "Hyperthyroidism," and "Other conditions." The classification task is simpler than that of the nonhierarchical network shown in Figure 4.12. The prior probability of each category is higher in this network.

The specialized network for hypothyroidism, shown in Figure 4.14, takes as inputs all patients that were classified in *hypothyroidism* in the triage network and discriminates the patterns of *primary hypothyroidism*, *secondary hypothyroidism*, *compensated hypothyroidism,* and *hypothyroidism not otherwise specified*.

Figure 4.14. A specialized neural networks for hypothyroidism.



This network has as inputs only the cases classified as "Hypothyroidism" by the triage network. It operates with fewer examples, and has a structure that is simpler than that of the generic network shown in Figure 4.12.

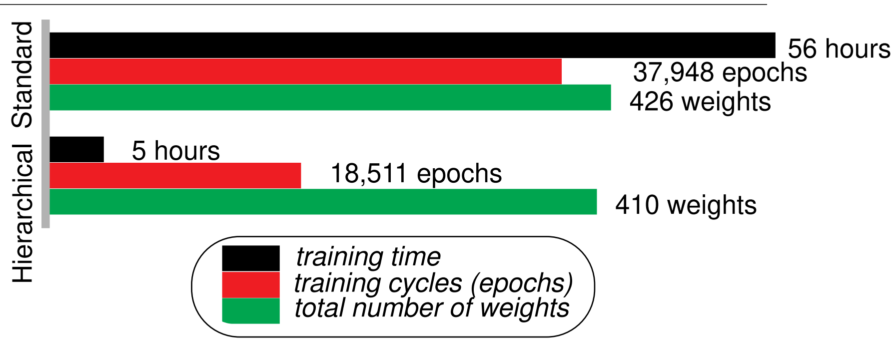
Table 4.4 shows the distribution of the output categories in the training set. Some patients had more than one diagnosis. Input attributes included age, gender, current medications, pregnancy status, previous thyroid surgery, presence of other illness, treatment with ^{131}I, clinical signs, and laboratory values for TSH, TT_4, T_4U, T_3, and TBG. Missing values were imputed as their means (in the case of continuous variables) or their mode (in the case of categorical variables).

Table 4.4. **Distribution of patterns.**

Output Category	Frequency	Percentage
Normal	6771	72.52
Hyperthyroidism, NOS	193	2.07
Primary hyperthyroidism	21	2.25×10^{-3}
Toxic goiter	18	1.93×10^{-3}
Secondary hyperthyroidism	9	9.64×10^{-4}
Hypothyroidism, NOS	1	1.07×10^{-4}
Primary hypothyroidism	239	2.56
Compensated hypothyroidism	419	4.49
Secondary hypothyroidism	8	8.57×10^{-4}
Other conditions	1658	17.76

The networks were trained as long as the error rate in the test set of 4586 patients was declining. When the error in the test set started to increase again, the stopping criterion was reached and training was discontinued. The networks were not trained up to convergence in order to avoid overfitting, as explained in Section 2.3.2. More details on an earlier implementation of HNN and the data set used for making the automated diagnosis of thyroid conditions can be found in [Ohno-Machado, 1994]. Figure 4.15 shows the time taken by the different systems to reach the stopping criterion.

Figure 4.15. Comparison of systems.



The time performance of hierarchical systems was clearly better. The perceptron was not able to discriminate rare patterns even after 4 x 10^5 epochs, indicating that the problem was probably nonlinearly separable.

Table 4.5 shows the sensitivities and specificities of the different systems after 90 minutes of training for the superclass (or intermediate abstraction) *Hypothyroidism*.[1]

Table 4.5. Prediction of class *Hypothyroidism.*

System	Sensitivity	Specificity	Epochs
Standard NN	49.25%	98.97%	650
Hierarchical NN	79.35%	98.82%	1,800

After approximately 90 minutes on an HP9000 workstation.

Table 4.6 shows the equivalent numbers for the class *compensated hypothyroidism*. These numbers are based on the test set. Note that the increase in sensitivity obtained by using HNNs is not coupled with a marked decrease in specificity. The superiority of the hierarchical system was clearly demonstrated in this complex problem. Not all possible subsets of variables were tried, but the results clearly confirm what was learned from the experiment using the artificial data set: HNNs can learn rare patterns faster than can their nonhierarchical counterparts. At any given point in time, classification performance of HNNs was better.

Table 4.6. Prediction of class *Compensated Hypothyroidism.*

System	Sensitivity	Specificity	Epochs
Standard NN	41.83%	98.45%	650
Hierarchical NN	65.87%	98.79%	3,800

After approximately 90 minutes on an HP9000 workstation.

4.6 Summary

Several hierarchical systems of neural networks have been proposed in the past. In bottom-up hierarchical designs, specialized neural networks are used to classify all inputs

1. This superclass contains all subtypes of hypothyroidism: *Hypothyroidism NOS*, *Primary Hypothyroidism*, *Compensated Hypothyroidism*, and *Secondary Hypothyroidism*.

according to one or more features. The results of these specialized networks are then used by an integrating network that provides the final solution. In top-down approaches, such as the one proposed in this work, a top-level network discriminates which inputs will be classified by each of the specialized networks, and the results of the latter will be the outputs of the system.

I built a top-down hierarchical system of neural networks to address the problem of rare-category recognition in backpropagation neural networks. I showed in two simplistic examples that HNNs can achieve the accuracy of nonhierarchical networks in shorter time, given approximately the same number of weights. In these examples, however, overfitting was not a problem and the networks were trained until they found the optimal classification. I also showed in a more complex example (in which controlling for overfitting was a concern) that HNNs can improve performance in a medical diagnostic task.

CHAPTER 5

Sequential Neural Networks for Prognosis

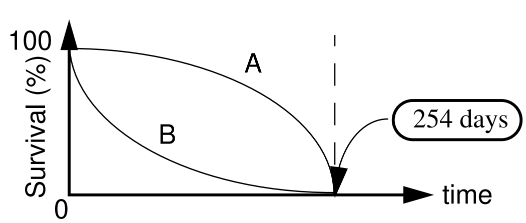
While diagnostic tasks usually require classification of cases for one time point, prognostic tasks may require the assessment of a patient's status over time. Sequential neural networks were built to determine whether the difficulties that standard neural networks have in detecting low-frequency patterns in prognostic tasks can be overcome. Section 5.1 describes the use of sequential neural networks for prognosis. Section 5.2 reviews the essential concepts in survival analysis and discusses existing methods, their advantages, and their deficiencies. Section 5.3 shows how sequential neural networks can be used as alternatives to current survival analysis models for establishing prognoses over time.

5.1 Sequential Neural Networks

Outcome prediction is not an easy task in medicine. Even though researchers are familiar with the concept of *survival in five years* for the study of outcomes of deadly diseases, there is an increasing need for predictions for shorter intervals of time, and for projections of patient-specific survival curves. Neural networks have been shown to be good predictors of outcomes in a variety of medical applications, but current neural network models often mimic the models of five-year survival (i.e., they make predictions for one specific time point) and fail to provide a complete picture of a temporal pattern of disease progression. A sequential system of neural networks can produce patient-specific survival curves and facilitate the recognition of temporal patterns of disease, making a distinction between patterns of fast and slow development. The accurate prediction of survival for different patients makes it easier to track the patients whose disease development is fast. These patients usually demand more aggressive management than those whose diseases exhibit patterns of slow development.

Current use of neural networks in survival analysis is aimed at producing either (1) an absolute estimate of survival (e.g., 254 plus or minus 10 days) or (2) a single point estimate of survival (e.g., 70 percent probability of survival in five years). An absolute estimate of survival may be useful for certain purposes, but provides no information on whether disease development seems to be fast or slow for a given patient. Figure 5.1 shows how an absolute estimate of survival of 254 days can be produced by different patterns of disease progression.

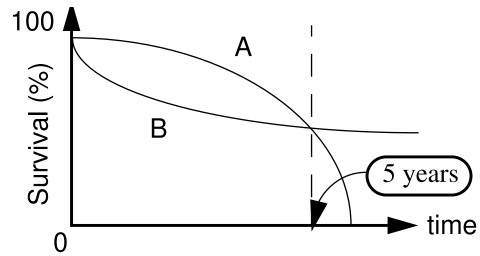
Figure 5.1. Absolute survival estimate.



Patients A and B have the same estimate of survival (254 days), but disease progression for B is faster in the beginning.

A single-point estimate cannot illustrate temporal patterns of disease development. For example, Figure 5.2 shows how a single point estimate may be misleading for long-term predictions.

Figure 5.2. Point estimated of survival.

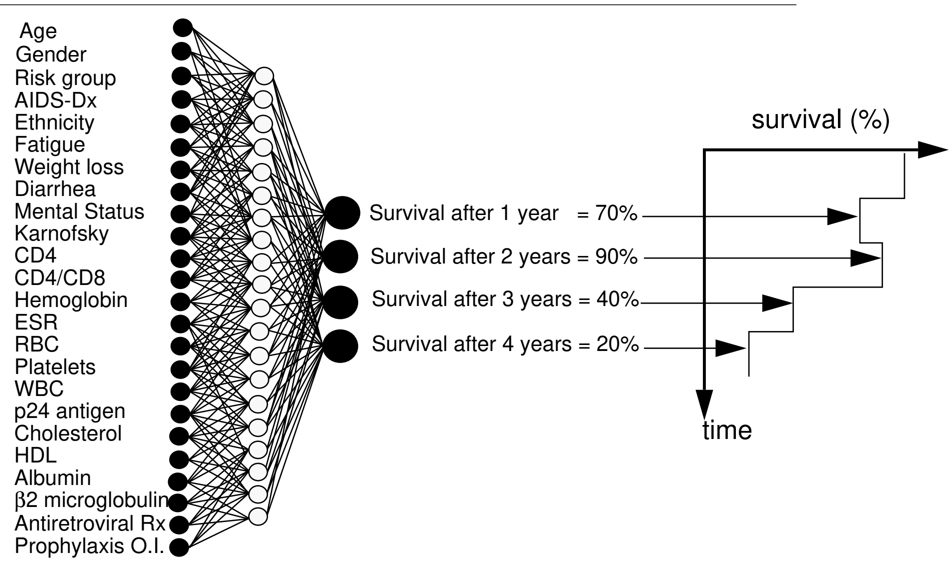


Patients A and B have the same probability of being dead in five years (50%), but the long-term prognosis for B is better.

5.1.1 Neural networks and survival curves

A nonsequential neural network that produces multiple point estimates of survival, such as the one shown in Figure 5.3, can be used to produce survival curves.

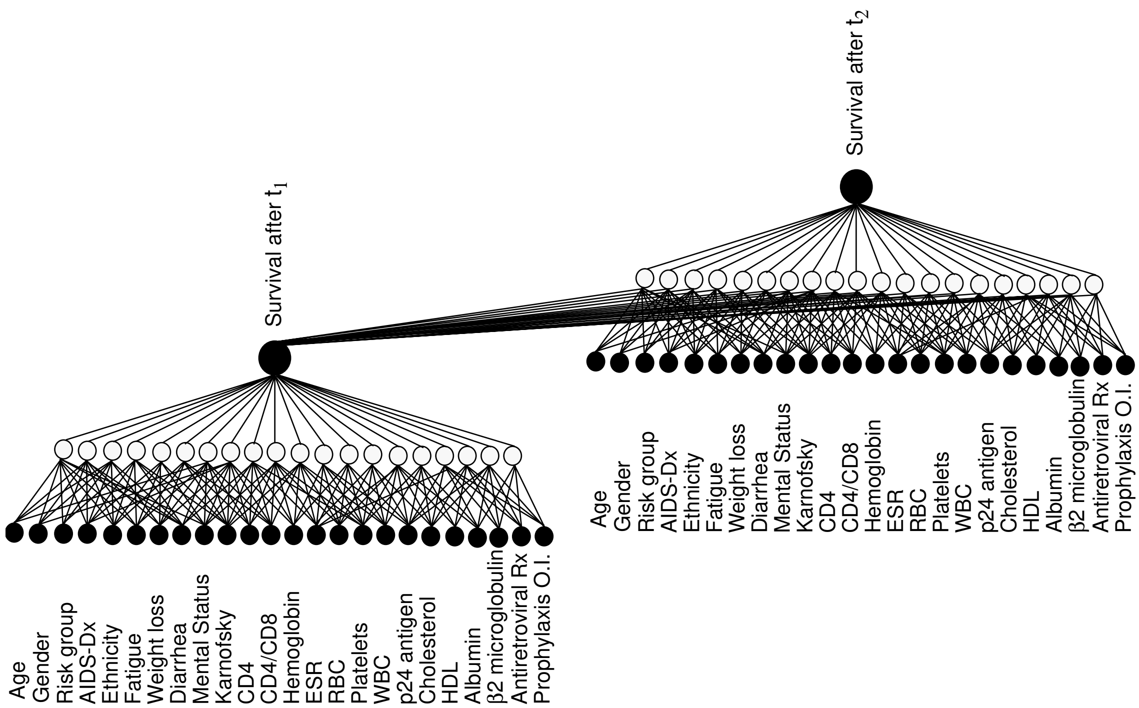
Figure 5.3. Standard neural network and nonmonotonic survival curve.



A neural network with multiple point estimates may produce a survival curve. However, predictions from one interval cannot enhance predictions for other intervals, and for a given patient an anomalous survival curve, such as the one shown here, may result.

This network, however, is not able to use predictions for a given interval to enhance predictions for another interval. For example, this network may estimate that survival in one year for a given patient is 70 percent and survival in two years for the same patient is 90 percent (which is impossible!) because it cannot take into account predictions for year 1 in the model that predicts survival in year 2. This network is also unable to deal with censored data. Figure 5.4 shows a sequential system of networks that can deal with these problems.

Figure 5.4. Sequential neural network models.



In a sequential system of neural networks, predictions for time t_1 are inputs to the model that makes predictions for time t_2, and so on. Anomalies in survival curves are minimized by utilizing information of some intervals as inputs.

High accuracy can be achieved with sequential neural networks, even when low-frequency patterns are present, as will be shown in Chapters 7 and 8. But their use has also some disadvantages. It implies the construction of more complex models that may require long training times. It may induce errors in predictions if the input predictions are too inaccurate, as will be discussed in Chapter 9.

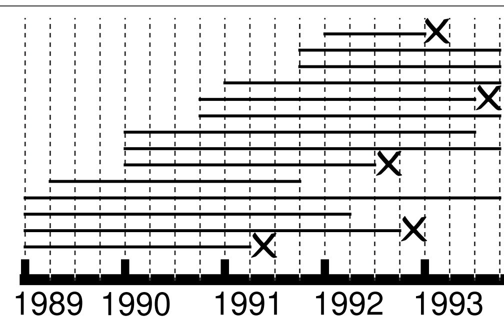
5.2 The Analysis of Survival Data

The understanding of disease progression is facilitated by survival analysis methods. In HIV infection, a relatively new disease, there is still much to be learned about patterns of disease progression. Existing survival analysis methods present some deficiencies, especially when data are censored and time-dependent variables are necessary.

5.2.1 Censored data

Survival analysis is confronted with several sources of difficulties. One of them is the possibility that some individuals may not be observed for the full time up to death [Cox, 1984]. This type of missing data is called censored data and is exemplified in Figure 5.5. For example, in a study to determine the five-year survival with AIDS, it may happen that some individuals are alive at the end of study, and although researchers know that these patients survived more than five years, they do not know exactly how long (Type I censoring) [Lee, 1992].

Figure 5.5. Example of censored data.



X indicates that the patient died during the study. Patients' data with no X were censored at time corresponding to the end of the segment. The end of the study is January 1994. Patients still alive at this date will have data corresponding to Type I censored data.

Another source of difficulty in survival analysis is the possibility that some patients are lost to follow-up in the middle of the study (Type II censoring). Patients may also enter the study at different times and their data may be censored by a combination of Type I and

Type II censoring (Type III censoring). Censoring of Types I and II is called singly censored data, and censoring of Type III is called progressively censored data. These three types of censoring are examples of *right censoring*. It also may be unclear when the patients entered the study (e.g., patients infected with HIV, where the date of infection is usually unclear). Data from these patients is considered *left censored*.

Using censored data is an important feature of a survival analysis method. Even though data is incomplete, it contains a certain amount of information. For example, if patients with AIDS are lost to follow-up after five years, their cases provide the important information that these patients have survived at least that long. Given the difficulties of obtaining and collecting information on a significant number of patients for clinical studies, it is evident that the amount of information lost should be minimal. Discarding censored cases from survival analysis studies should be avoided whenever possible. As we will see later, however, certain methods of survival analysis have difficulties in dealing with censored data.

5.2.2 Functions of survival time

Two functions of central interest in modeling survival data are the survivor function and the hazard function. The survivor function is defined as the probability that an individual survives at least up to a certain time. The hazard function is defines as the probability that an individual will die at a certain time, conditioned on his survival up to that time, and denotes the instantaneous death rate [Collet, 1994]. Survival analysis models, such as actuarial life tables [Cutler, 1958], product-limit estimators [Kaplan, 1958], proportional hazards [Cox, 1984], and fully parametric models [Lee, 1992] produce estimates for both the survivor function and the hazard function. Parametric methods of survival analysis require specification of a probability density function for estimating these functions. Non-parametric models do not require this specification, and are predominant in the biomedical literature. The process of searching for an appropriate model of distribution for parametric models may be too time-consuming or economically impractical. In that case, researchers may use nonparametric models, such as regression trees [Segal, 1989] and neural networks [Ravdin, 1992].

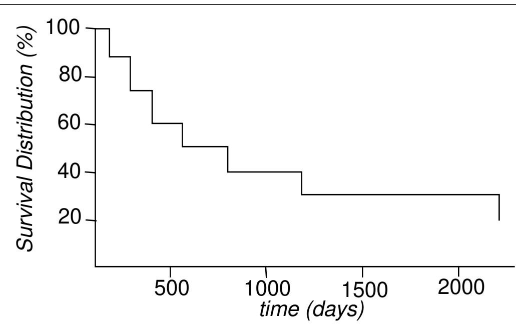
5.2.3 Life tables and product-limit estimators

Actuarial life tables and product-limit estimators of survivor functions (also known as Kaplan-Meier survival curves) are simple nonparametric models that help researchers to summarize survival data. Both types of models involve the assumption that the survival of an individual at time *t* is conditioned on his survival at time *t-1*. The survivor function for actuarial life tables and product-limit estimators is

$$S(t) = \Pi\left(\frac{n_j - d_j}{n_j}\right)$$

where d_j is the number of deaths in interval *j* and n_j is the number of individuals at risk [Collet, 1994]. In the actuarial method, n_j is the *average* number of individuals at risk. Actuarial life tables and product-limit estimators differ in the way that time intervals are built. The former model predefines intervals of equal duration and groups deaths in those intervals. The latter model builds one interval for each death, and therefore does not cause loss of information. An example of a Kaplan-Meier survival curve is shown in Figure 5.6. Both models allow for censored data. Neural network models, as we will demonstrate, also can estimate survivor functions in intervals when censored data are present.

Figure 5.6. Kaplan–Meier survival curve.



Survival curve for 428 AIDS patients of the ATHOS data set.

Life tables and product-limit estimators are good for describing survival of a group of individuals. Comparison of different survival curves produced by these methods is done using log-rank [Tarone, 1979] or Breslow tests [Breslow, 1970]. Survival curves can be built for a group of patients who share a given characteristic. For example, patients who are over 40 years of age, and compared to survival curves for younger patients. Continuous variables have to be discretized, and a threshold value needs to be arbitrarily chosen to allow the division of patients into major groups. It is not possible to create individualized survival curves for each patient. Although comparisons with multiple variables are possible (e.g., age and gender), the number of patients in each resulting group must not fall below a certain minimum, otherwise a reliable survival curve cannot be created. These models are good for explaining existing data and making univariate or simple multivariate comparisons, but their prognostic use is limited.

5.2.4 Parametric models for disease progression

Parametric models of disease progression, such as the accelerated failure-times model, have also been used for survival analysis, where the restrictive assumption of a fixed distribution is traded for efficiency and facility of mathematical manipulation. The assumption of a fixed distribution sometimes allows for an analytical solution to a survival analysis problem, and helps to minimize the problem of overfitting the data. Small differences in goodness-of-fit can often be detected when parametric models are used. However, the choice of the function that represents survival in these methods is arbitrary. If different groups of patients happen to have survival curves that do not fit a given distribution, the models have limited value. Nonparametric methods have the advantage of not being limited to a certain class of functions.

5.2.5 Survival analysis for prognosis

The models mentioned so far are usually employed to explain the data, rather than to make predictions. An important requirement for a classifier is its ability to use as much information as it can on an individual case to make accurate predictions. To determine the predictive value of different variables for the progression of disease, a logistic regression model, discussed in Chapter 7, or a Cox proportional hazards models [Cox, 1972] can be used. In the domain of HIV infection, the most commonly used method is the latter.

When the task is to define which variables (or cofactors) influence the survival (and therefore allow an individualized prediction of survival) other methods, shown in Section 5.2.6, are better suited.

5.2.6 Cox proportional hazards and other logistic regression models

The Cox proportional hazards model is frequently used to study the importance of covariates for survival and to produce survival prognoses. The Cox model is a multiple regression semi-parametric model that allows modeling of continuous covariates. It requires the assumption that there is a simplifying transformation of the initial data and that the hazards for the different individuals are proportional. A baseline hazard has to be estimated and hazards for individuals are multiples of the baseline hazard. Although there are methods to test the validity of the assumptions [Chale, 1992], they are seldom used in practice. For example, Hanson et al. described a cohort of HIV+ patients for whom the assumption of hazard proportionality does not hold [1993]. For these cases, fully nonparametric models of survival that allow non-proportional hazards are needed. Actuarial life tables and product-limit estimators could be used to produce survivor estimates for different strata, which should then be compared by the Mantel-Haenszel method [1959], but this method cannot deal with continuous variables. Furthermore, stratifying data may result in few patients in each stratum, decreasing the statistical significance of the study. Other regression models, in which the proportionality assumption is not required, have been used. In the pooled logistic regression model, discussed in Chapter 7, data obtained from one individual may be used more than once as inputs to the logistic regression. For example, if data from a patient was collected at intervals I_1, I_2, and I_3, the researcher may use the three different records as inputs to the logistic regression models.

The use of time-dependent variables is important in some survival analysis models. Time-dependent variables, as the name suggests, are the ones that are allowed to change over time. For example, initial values for cholesterol may be of great importance in determining development of coronary heart disease. If cholesterol levels are entered in a model as time-dependent variables, the *evolution* of cholesterol levels can be modeled, so initial cholesterol levels, as well as those measured at the end of the first and second intervals, may be used as prognostic variables. Obviously, there are other variables such as “Gender” and “Ethnic origin” that cannot need to be modeled as time-dependent variables.

Neural networks have not been used extensively for survival analysis, but they offer no conceptual obstacle to handling censored data or time-dependent variables. Sequential neural networks can easily deal with censored data. Time-dependent variables are more easily handled in sequential rather than standard neural networks. The summary of other authors’ experiences with neural networks for survival analysis is presented next.

5.2.7 Previous work on neural networks for survival analysis

The most common applications of neural networks in clinical medicine have been for diagnosis of diseases. A few reports on the use of neural networks for other kinds of classification tasks have been published. These systems determine prognosis after cardiopulmonary resuscitation [Ebell, 1993], strategies for weaning from respiratory support [Ashutosh, 1992], tumor stage in oncology [Burke, 1994], graft outcome after liver transplantation [Doyle, 1994], and prognosis in trauma [McGonigal, 1993]. Ravdin et al. [1992] studied the prognosis of breast cancer patients using neural network models. In their models, time was entered as a predictor, and each patient had as many entries in the model as the number of intervals during which she was alive. The intervals were derived from Kaplan–Meier estimates, according to percentiles. The survival curves for the quartiles that generated similar prognoses were then plotted, and were compared with those of a Cox regression. There were no significant differences. The data probably respected the proportionality assumption, although this hypothesis was not tested explicitly. A bias was introduced by coding time as a covariate (so that no time-dependent variables were used), and the authors had to balance the input data set to account for that bias. The importance of individual variables to the overall prediction of survival was not analyzed. The work of Ravdin et al. [1993] was one of the first studies to address the use of neural networks for survival analysis using real clinical data, producing accurate estimates for survival of breast cancer patients, and raising the important issue as to how to deal with censored data in neural network implementations for survival analysis. McGonigal et al. [1993] have applied neural network models to assess the probability of survival for trauma patients, and the neural network model compared favorably with other systems. Both these neural network models for survival analysis were implemented nonsequentially.

5.3 Survival Analysis Using the Standard and the Sequential Methods

The decomposition of a classification task into small subcomponents facilitates learning in neural network systems, especially when there are sequences of solutions and the frequency (prior probability) of certain categories is low. This approach can be used in survival analysis, where the frequency of events in each interval may be low compared to the overall number of individuals. The number of options for building a sequence of abstractions to use in sequential neural networks that perform survival analysis can be considered smaller than the number of options available in other classification problems, since the abstractions are all based in one single concept: time. Nevertheless, by considering survival analysis a type of classification problem that is amenable to decomposition by sequential neural networks, and in which survival times are classified according to predefined intervals, I have to introduce the assumption that no significant information is lost by transforming a real number that represents the exact survival time of an individual (e.g., 58.4 days) into an integer that represents the interval in which that survival time is contained (e.g., 2 months). Although in survival analysis the choice of intermediate abstractions is not as varied as in other classification problems, some principles for qualitatively modeling classification tasks into useful hierarchies and the development of requirements for building sequential neural networks for survival analysis are important contributions to this work. Before describing the experiments with sequential neural networks in Chapters 7 and 8, some definitions are needed:

Standard model: A model that predicts outcomes in a single interval or a single point in time, using no information on other intervals.

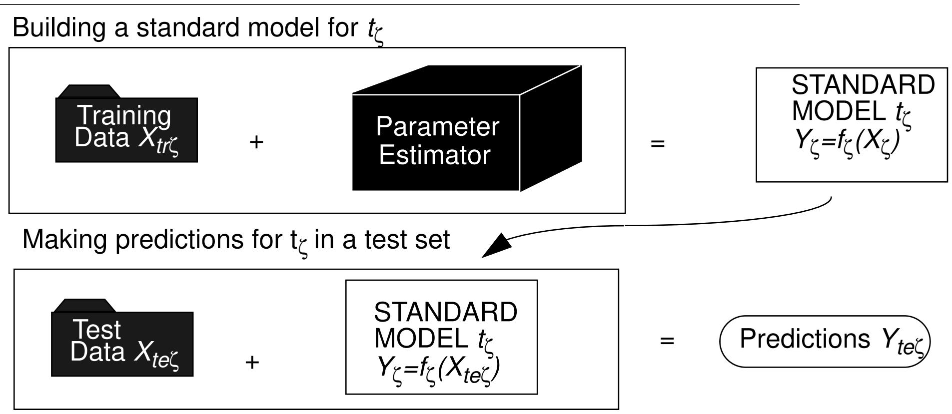
Sequential model: A model that predicts outcomes in a single interval or a single point in time using predictions for other intervals (among other variables).

Informative Interval: An interval or point in time for which predictions are made using a standard model. Its predictions can be used as input, along with other variables, to a sequential model.

Informed Interval: An interval or point in time for which predictions are made using a sequential model.

An example of a standard model is shown in Figure 5.7.

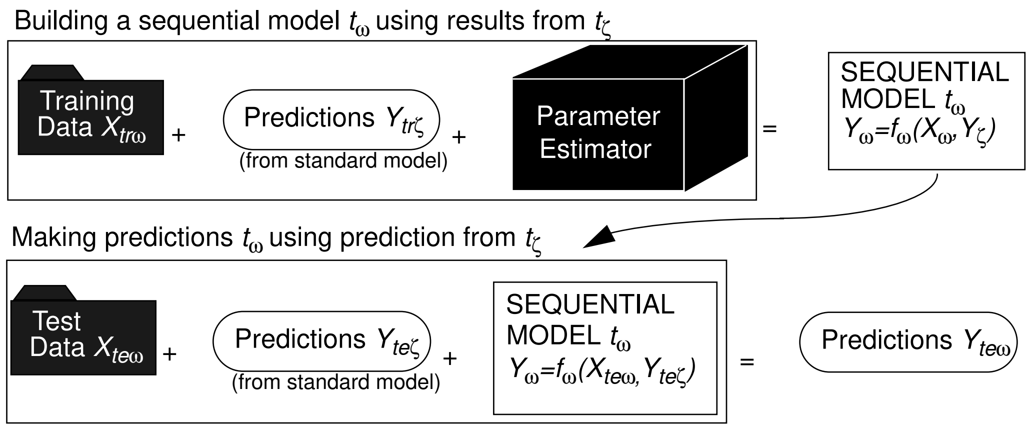
Figure 5.7. Standard model.



Standard models can be constructed for neural networks and other classification models. In a standard model, a simple model is created for each interval. For example, a simple logistic regression model for interval ζ (e.g., 2 years) can be created by providing training data $X_{tr\zeta}$ to a parameter estimator (e.g., SAS procedure LOGISTIC). The standard model for interval ζ is the mapping of X_ζ to Y_ζ, defined as the function f_ζ. This function can be applied to both training and test data ($X_{tr\zeta}$ and $X_{te\zeta}$, respectively) to produce predictions for both sets (e.g., 70 percent probability of survival in 2 years). Predictions for other intervals are NOT entered as inputs.

In the sequential model, predictions from a given interval are entered in a model that makes predictions for another interval, as shown in Figure 5.8. The number of cases used for sequential models is usually limited by the minimum number of noncensored cases in the intervals considered, which is usually the number of cases available for prediction in the longest interval. For example, if predictions for year 8 are entered in a model that predicts CHD in year 4, only the cases who had follow-up of at least 8 years can be used. Evidently, if predictions for year 4 were entered in a model that predicts CHD in year 8, only the cases who were followed for at least 8 years can be used.

Figure 5.8. Sequential model.



Standard models can be constructed for neural networks and other classification models. In a sequential model, a complex model is created for each interval, using predictions from another interval. For example, a sequential logistic regression model for interval ω (e.g., 4 years) can be created by providing training data $X_{tr\omega}$ to a parameter estimator (e.g., SAS/STAT procedure LOGISTIC) and predictions $Y_{tr\zeta}$ for another interval ζ (e.g., 2 years), obtained from the standard model for interval ζ. The sequential model for interval ω is the mapping of X_ω and Y_ζ to Y_ω, defined as the function f_ω. This function can be applied to both training and test data ($X_{tr\omega}$ and $X_{te\omega}$, respectively) to produce predictions for both sets. Sequential models can be built in either ascending order (i.e., ζ precedes ω) or descending order (i.e., ω precedes ζ).

Chapters 7 and 8 provide many examples of sequential models.

5.4 Summary

Medical researchers who perform prognostic modeling usually oversimplify the problem by choosing a single point in time to predict outcomes (e.g., death in five years). This approach not only fails to differentiate patterns of disease progression, but also wastes important information that is usually available in time-oriented research data bases. The adequate use of time-oriented data bases can improve the performance of prognostic systems if the interdependencies among prognoses at different intervals of time are explicitly modeled. In such models, predictions for a certain interval of time (e.g., death within one year) are influenced by predictions made for other intervals, and prognostic survival curves that provide consistent estimates for several points in time can be produced. The recognition of temporal patterns by neural networks can be facilitated with a sequential system. In this system, predictions from time t_ω can be improved if predictions on time t_ζ are provided as inputs, and so on. A temporal pattern can be delineated this way, even for rare cases. Current survival analysis models require assumptions that may not always be verified in real-world data. An introduction to these models was given in this chapter, and the need for sequential systems of neural networks in survival analysis was justified. A system of neural networks constructed sequentially can make prognoses accurately, producing survival curves that define specific temporal patterns of disease progression. Sequential versions of other classification models, such as logistic regression models, can also be built.

CHAPTER 6 Evaluation Methods

In this chapter, I will explain the evaluation methods that are currently used to assess binary classification tasks, and discuss the rationale for choosing certain ones to evaluate my experiments. Performances of classification models should be compared not only on the basis of the number of correctly classified cases, which is dependent on the number of cases per category, but also on the basis of how much and which type of information was actually gained by using a certain model. The communication of results has to be made in a language that is understandable by the average researcher, who is often a specialist in a medical domain but not a specialist in evaluation methods.

There are currently no universally accepted guidelines on how to evaluate the performance of different neural network models, except the reduction in entropy, the comparison of areas under the ROC curve (see Section 2.3), and the total number of correct answers provided by the model when applied to a test set, which in fact represents the *generalization* capability of neural network models. For regression models, measurements that represent the fitness of the data to the model are obtained from the training data and may guide the selection of variables.

6.1 Evaluation of a Model's Goodness-of-Fit

The reduction in error accounted for the inclusion of variables in regression models can be evaluated by the Akaike's Information Criterion (AIC) [Akaike, 1977]. The AIC is based on the maximum likelihood test statistic for testing model X, with r degrees of freedom, against the saturated model, with q degrees of freedom, and corresponds to

$$A_X = G^2(X) - [q - r]$$

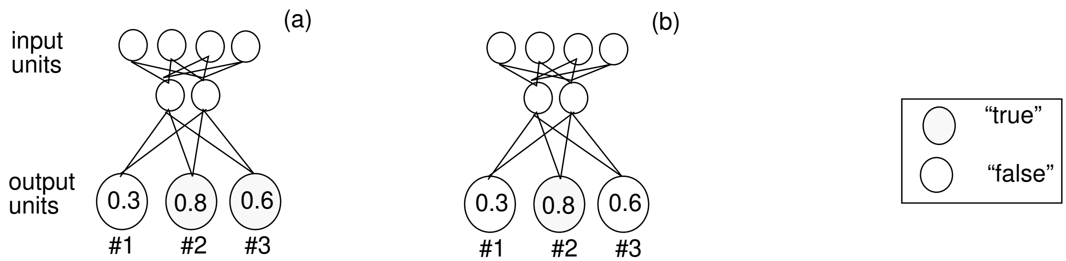
for log-linear models, where G^2 is the likelihood ratio test statistic [Christensen, 1990]. The AIC is usually used to compare regression models that differ in the number of variables, and by itself provides no insight on how accurate a model is.

There are currently no tests for assessing the goodness-of-fit of neural network models that are adjusted for the number of parameters. There are, however, several tests that assess predictive performance on test sets and that can also provide some insight into how accurate (and therefore useful) the models are. These tests will be explained next.

6.2 Evaluation of Continuous Estimates of Binary Outcomes

Classifiers such as neural networks usually produce real numbers as their outputs. These real numbers may be processed or interpreted in different ways to provide a classification: the researcher may establish a threshold above which the final output of a certain unit will be "1" or "true," or he or she may choose as correct the output that has the highest number in a set of mutually exclusive alternatives (see Figure 6.1). When the actual outcome is binary (like the one in our survival analysis models, where a given individual is either dead or alive at a given interval) and the estimates, or predictions, are given in real numbers, the evaluation of the model requires that two aspects of the estimates be evaluated: calibration and resolution.

Figure 6.1. Interpretation of continuous estimates from a neural network.



(a) If a threshold of 0.5 is chosen, Outputs #2 and #3 are considered “true.” (b) Output #2 is considered “true” if the output with the highest number is chosen.

6.2.1 Example

In order to illustrate some of the statistics explained in this chapter, I will use a very simple example. Suppose the predictions of a predictive survival analysis are the ones shown in Table 6.1.

Table 6.1. **Simple example of outcomes and predictions.**

Case number	Outcome*	Prediction
1	1	0.9
2	0	0.2
3	1	0.6
4	0	0.1
5	0	0.5
6	0	0.4
7	1	0.8
8	1	0.7
9	1	0.4
10	0	0.6

*Real status according to gold standard, “1” = dead and “0” = alive

(1) How close are the predictions to the real outcomes?

(2) Are the predictions systematically high or low?

(3) How much do predictions for cases with real outcome “1” differ from those with real outcome “0”?

In order to answer these questions, the researcher can utilize several evaluation methods. I will comment on the most frequently used evaluation methods next, discussing briefly their advantages and disadvantages.

6.2.2 Brier score

A simple way to evaluate how much the predictions departed from the outcomes in our example is to calculate the difference between each prediction and each outcome. In our example, the differences would be 0.9 - 1 = - 0.1 for the first row in Table 6.1, 0.2 - 0 = 0.2 for the second row, and so on. Assuming that the penalty for type I and type II errors is the same (i.e., that it is as bad to state that someone will die when in fact he survives as it is to state that someone will survive when in fact he dies), we can square the errors and get to a global error by summing the squares for all individuals, as shown in Table 6.2. The perfect classifier would produce predictions that correspond to zero global error. The Brier score, or probability score, is the global error divided by the number of cases, and represents a type of average error per case.

Table 6.2. Errors for each case and global error.

Case #	Outcome*	Prediction	Error	Square of Error
1	1	0.9	-0.1	0.01
2	0	0.2	0.2	0.04
3	1	0.6	-0.4	0.16
4	0	0.1	0.1	0.01
5	0	0.5	0.5	0.25
6	0	0.4	0.4	0.16
7	1	0.8	-0.2	0.04
8	1	0.7	-0.3	0.09
9	1	0.4	-0.6	0.36
10	0	0.6	0.6	0.36
				Global error = 1.48

* Real status according to gold standard, “1” = dead and “0” = alive

Although there are methods to compare Brier scores from different models [Redelmeier, 1991], they do not provide much insight into the prediction performance and do not answer question 2 (*Are the predictions systematically high or low?*) and 3 (*How much do predictions for cases with outcome "1" differ from those with outcome "0"?*) from above. In order to answer those question, the Brier score has been decomposed into more interpretable components by many authors. The most popular decomposition is attributed to Yates [Yates, 1982]. Calibration and resolution are the most important components of the Brier score and will be discussed next. All components can be visualized in a covariance graph, which will be discussed in Section 6.5.3.

6.3 Calibration

Calibration is a measure of how close the predictions of a given model are to the real outcome. It indicates whether the predictions are high or low when compared to the real outcomes.

6.3.1 Calibration-in-the-large

One of the ways to assess calibration is to take the difference between the average observation and the average outcome of a given group. If an outcome of "1" is a desirable outcome, this statistic can be viewed as a measure of "optimism." For example, if the average estimated prediction of survival produced by a model for a given group of patients is 0.85 and the average survival of the patients is 0.70, the model is being too optimistic on average. This type of calibration is called "calibration-in-the large," because it takes into account the whole test sample, and is simple to calculate and interpret. This measure answers question (2) from the previous section (*Are the predictions systematically high or low?*), but it is not very useful in practice, since there are many examples in which a model can be perfectly calibrated-in-the-large, and yet provide no information.

Consider an example where a physician is asked to assess the probability that an individual survives in an ICU. If the physician is not provided with any information about the patient (e.g., gender, age, disease, vital signs), his best guess will be the prior probability of a patient surviving in that unit. For example, if he knows that 70 percent of the patients survive in that unit, he should guess 0.7 and would indeed be perfectly calibrated according to this statistic. Note that a model that always produces as its prediction the prior probability of the event (like our physician) will always be perfectly "calibrated-in-the-large," but may never discriminate patients who survive from patients who die. In other examples, each individual prediction could be very far from the real outcome, but the errors might compensate each other and the final average number could falsely represent excellent calibration.

6.3.2 Calibration-in-the-small

A more refined way to measure calibration, called "calibration-in-the-small," is done by dividing the sample into smaller groups sorted by predictions, calculating the sum of predictions and sum of outcomes for each group, and determining whether there are any statistically significant differences between the expected and observed numbers by a simple χ^2 method. In practice, this procedure is done by sorting the predictions, choosing a number *n*, dividing the sorted set uniformly into *n* groups, summing predictions and summing outcomes for each group, and performing a χ^2 test with *n-2* degrees of freedom. Using the example of Section 6.2.1, three groups could be determined, as shown in Table 6.3.

Table 6.3. **Cases from Table 1 sorted by prediction.**

Case #	Outcome*	Prediction	Group
4	0	0.1	1
2	0	0.2	
6	0	0.4	
group sum	0	0.7	
9	1	0.4	2
5	0	0.5	
10	0	0.6	
group sum	1	1.5	
3	1	0.6	3
8	1	0.7	
7	1	0.8	
1	1	0.9	
group sum	4	3	

* Real status according to gold standard, “1” = dead and “0” = alive

A χ^2 test with 1 degree of freedom that resulted in a $p > 0.05$ would indicate that there are no statistically significant differences between summed predictions and summed outcomes, and therefore there is good calibration[1]. This test is usually referred to as the Hosmer-Lemeshow test and is a common measure of goodness-of-fit for logistic regression models [Glantz, 1990]. Cox proposed a similar method to assess the agreement between a sequence of observations and a set of probabilities, mainly for use in the analysis of logistic regression models [Cox, 1984], but suggested some additional statistics when small numbers of data are available. The calculation and interpretation of these statistics is not as easy as the ones shown in this chapter, and they are rarely used in practice.

The Hosmer-Lemeshow test, as opposed to other methods to assess calibration, such as the graphical methods presented later in this chapter, has the advantage that it can

[1] This test could not be done with the simple Table 6.3, since the counts in the cells are below five, and there is an expected value of zero in the first group. Alternative tests to be used in situations such as this one are discussed in [Freeman, 1987].

determine to a certain degree of confidence whether there are statistical differences between the predictions and the outcomes. Graphical methods, as we will see later, are useful to detect gross calibration deviances, but are not adequate to detect small deviances.

6.4 Resolution

Resolution, or discrimination, measures how much the model is able to separate cases with outcome "1" from those with outcome "0." It is independent of calibration, since there are models that have good calibration but poor resolution and models that have good resolution but poor calibration.

6.4.1 Slope

Slope is a simple measure of resolution that represents the difference between the average prediction y_1 for cases with outcome "1" and the average prediction y_0 for cases with outcome "0." A slope of "0" would indicate no resolution, whereas a slope of "1" would indicate perfect resolution on average. The slope is calculated by simply dividing the sample into two groups according to their outcome and calculating the difference between the average prediction and the average outcome, as shown in Table 6.4.

Table 6.4. **Cases from Table 6.1 sorted by expected outcomes.**

Case #	Outcome*	Prediction
1	1	0.9
3	1	0.6
7	1	0.8
8	1	0.7
9	1	0.4
		average: 0.68
2	0	0.2
4	0	0.1
5	0	0.5
6	0	0.4
10	0	0.6
		average: 0.36

*Real status according to gold standard, “1” = dead and “0” = alive

The slope in this case is 0.68 - 0.36 = 0.32. Although the slope is easy to calculate, it is not easy to interpret nor useful for comparisons across different models. It cannot determine to a certain degree of confidence whether a model’s predictions are discriminatory or not, and is used mainly for illustrative purposes when there is a gross deviance in resolution.

6.4.2 Pairwise discrimination

In pairwise discrimination, resolution is assessed by (a) making all possible pairs of cases in which one case represents outcome “1” and the other represents outcome “0,” (b) counting the number of concordant pairs (i.e., pairs where the prediction for the case with outcome “1” is higher than that of the case with outcome “0”), and (c) scaling the sum according to the total number of pairs, as shown in Table 6.5. When the number of concordant pairs is high compared to the total number of pairs, there is good discrimination between cases with outcome “0” and those with outcome “1.”

Table 6.5. All possible pairs composed of one case with outcome "1" and one case with outcome "0."

Pair case# - case#	Prediction 1st case φ(#1)	Prediction 2nd case φ(#2)	φ(#1)>φ(#2) Concordant	φ(#1)<φ(#2) Discordant	φ(#1) = φ(#2) Tie
1 - 2	0.9	0.2	Y		
1 - 4	0.9	0.1	Y		
1 - 5	0.9	0.5	Y		
1 - 6	0.9	0.4	Y		
1 - 10	0.9	0.6	Y		
3 - 2	0.6	0.2	Y		
3 - 4	0.6	0.1	Y		
3 - 5	0.6	0.5	Y		
3 - 6	0.6	0.4	Y		
3 - 10	0.6	0.6			Y
7 - 2	0.8	0.2	Y		
7 - 4	0.8	0.1	Y		
7 - 5	0.8	0.5	Y		
7 - 6	0.8	0.4	Y		
7 - 10	0.8	0.6	Y		
8 - 2	0.7	0.2	Y		
8 - 4	0.7	0.1	Y		
8 - 5	0.7	0.5	Y		
8 - 6	0.7	0.4	Y		
8 - 10	0.7	0.6	Y		
9 - 2	0.4	0.2	Y		
9 - 4	0.4	0.1	Y		
9 - 5	0.4	0.5		Y	
9 - 6	0.4	0.4			Y
9 - 10	0.4	0.6		Y	
Total			21	2	2
c-index = [21 + (2/2)]/25 = 0.88					

* Real status according to gold standard, "1" = dead and "0" = alive

There are several pairwise discrimination indices, which vary basically in the way they handle the ties (i.e., pairs that have exactly the same prediction for the two elements and are consequently neither concordant nor discordant).

The c-index is the index of pairwise discrimination presented above where the assumption is that one half of the ties accounts for concordant pairs and the other half accounts for discordant pairs [SAS, 1990]. It is calculated as

$$(Q + T/2) / n = Q / n_0 * n_1$$

where Q is the number of concordant pairs, T is the number of ties, n is the total number of pairs, n_0 is the number of cases with outcome "0," and n_1 is the number of cases with outcome "1." A c-index of "1" would mean perfect discrimination, whereas a c-index of 0.5 would mean no discrimination. Other frequently used measures of concordance of pairs are the Kendall tau-b and the Sommer's index [SAS, 1990]. The c-index is the most frequently used and easiest to interpret index. The c-index is also known as the Wilcoxon statistic [Larsen, 1990], for which standard errors can be calculated. Statistical tests of resolution using the Wilcoxon statistic can determine to a certain degree of confidence whether predictions from different models differ. Readers who are familiar with concepts such as test sensitivity, specificity, and ROC curves should not have any problems interpreting the Wilcoxon statistic, since it is synonymous to the area under the ROC curve, as shown by McNeil [1984] (see Section 6.5.2).

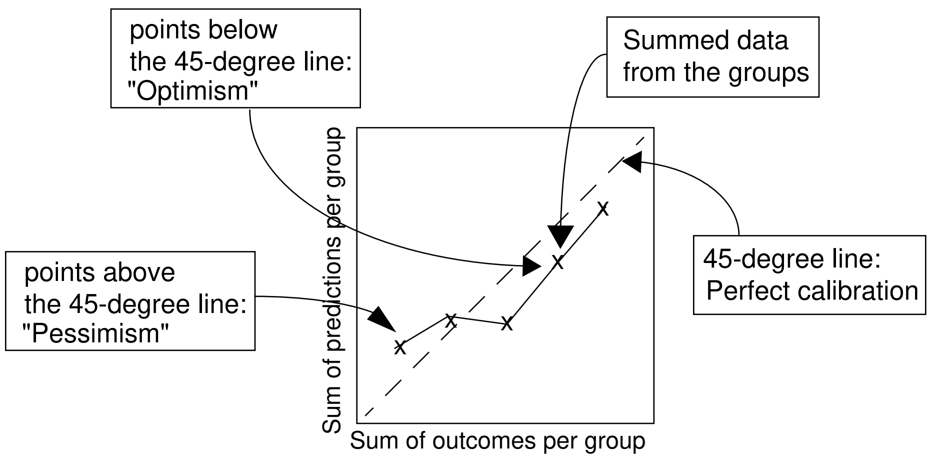
6.5 Graphical Methods

Graphical methods not only facilitate visualization of the results, but also can provide a powerful tool to detect gross discrepancies in calibration and resolution. Calibration can be visualized using calibration plots. Resolution can be visualized by ROC curves. A covariance graph can represent measures of calibration and resolution. Each of these graphical methods will be discussed next.

6.5.1 Calibration plot

A calibration plot is obtained by plotting the summed group predictions and summed group outcomes used to implement the Hosmer-Lemeshow test, discussed previously in Section 6.3.2. “Optimism” or “pessimism” can be detected in the plot if the line connecting the points is below or above the 45-degree line, as shown in Figure 6.2.

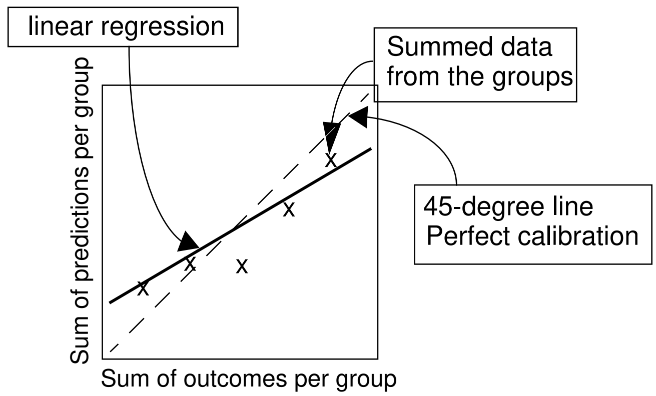
Figure 6.2. Example of a calibration plot.



If the outcome “1” is considered undesirable (e.g., death), then the points below the 45 degree line represent the model’s bias towards being optimistic (i.e., the model is predicting lower probabilities of dying in those points).

In another graphical representation of this idea, some authors suggested that sums of the predictions for the groups could be plotted against the sums of the outcomes, and the slope of the curve obtained by linear regression on these numbers compared to the slope of 1 (45-degree line), as shown in Figure 6.3. The equality of two slopes resulting from different models can be tested to assess differences in calibration [Larsen, 1990].

Figure 6.3. Example of linear regression analysis of group data.



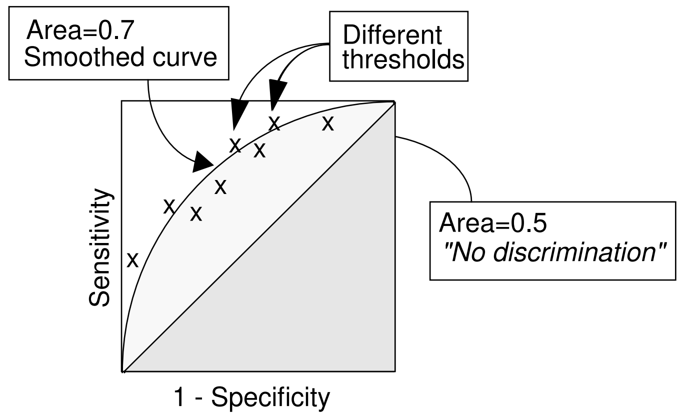
A linear regression is performed with the summed data from the groups. The regression line should not be very distant from the 45-degree line if the predictions are calibrated.

6.5.2 ROC curve

The ROC curve is a graphical representation of resolution. The ROC curve is a plot of the sensitivity versus one minus the specificity of a model in a binary classification task [Bernstein, 1988; Swets 1982; Egan 1975]. Sensitivity is defined as the number of correctly classified cases with outcome “1” (true positives) divided by the total number of cases with outcome “1” (real positives). Specificity is defined as the number of correctly classified cases with outcome “0” (true negatives) divided by the total number of cases with outcome “0” (real negatives). Each point in the ROC curve corresponds to a numeric threshold above which cases are classified as having outcome “1” (positive). At each point it is possible to define a 2 x 2 table with true positives, false positives, true negatives, and false negatives and hence plot sensitivity and specificity. The points defined in this plot can be approximated by a continuous function in a variety of ways, and therefore an area under the curve can be calculated. The easiest way is to connect the points with straight lines and calculate the areas of the resulting trapezoids [Centor, 1991].

The area under the ROC curve represents the discriminatory ability of the model. An area of 0.5, corresponding to the 45-degree line, as shown in Figure 6.4, represents no discriminatory ability, whereas an area of 1 represents perfect discriminatory ability.

Figure 6.4. Example of an ROC curve.



An area of 0.5 represents no resolution. An area of 1 represents perfect resolution.

Hanley and McNeil have shown that the area under the ROC curve calculated by the trapezoidal method corresponds to the Wilcoxon statistic. They have also shown ways to calculate the standard error and to compare resolution of different models using that statistic [Hanley, 1982 and 1983].

The comparison of areas under the ROC curve [Swets, 1982] implies that the researcher is interested in all range of cut-off values for deciding in favor of a classification, which rarely happens in practice. Hilden [1991] has criticized the indiscriminate use of ROC areas to compare diagnostic tests. It is sometimes more interesting to compare only portions of, rather than the entire, ROC curve, or to utilize other measures of performance when ROC curves cross [Moise, 1988]. The accuracy index may be based on just one ROC point. In this case, a value that corresponds to an acceptable specificity may be chosen, and the sensitivities may be compared, as suggested by McNeil [1984]. If necessary, other indexes [Centor, 1991] may be used.

6.5.3 Covariance graph

The covariance graph can represent measures of calibration, resolution, and scatter that are referred to in the decomposition of the Brier score [Arkes, 1995]. I will describe each component of the graph: bias, slope, and scatter.

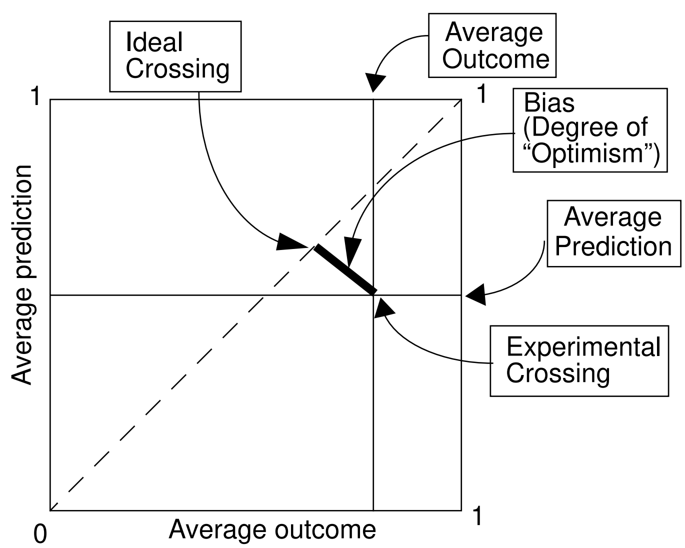
Bias is a measure of calibration-in-the-large, and is calculated by taking the square of the difference between the average prediction and the average outcome,

$bias = \Sigma(e_i - o_i)^2$

where i is the number of the case, e_i is the average prediction produced by the model for case i, and o_i is the average outcome for case i.

This difference is represented in the covariance graph by the distance between the crossing of the lines representing the average prediction and the average outcome and the 45-degree line, as shown in Figure 6.5. If the crossing happens below the 45-degree line, and outcome "1" is not desirable, then the predictions are "optimistic." For example, suppose the outcome is "0" if a person is alive and "1" if a person is dead. If the crossing of the average prediction and the average outcome occurs below the 45-degree line, then the predictions are too optimistic: they indicate that the average of the probabilities of being dead as given by the model are below the ones that actually could be verified.

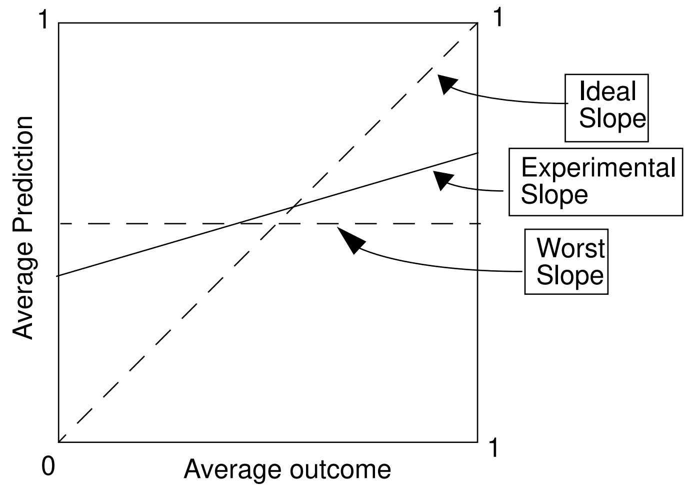
Figure 6.5. Assessing calibration in a covariance graph.



Calibration-in-the-large can be visualized by the bias.

A rough measure of resolution is represented in the covariance graph by the *slope,* discussed in Section 6.4.1. A perfect slope would correspond to 1, and would be represented by a 45-degree line in the graph, as shown in Figure 6.6.

Figure 6.6. Assessing resolution in a covariance graph.



Average prediction for cases with outcome “0” and average prediction for cases with outcome “1” are connected by a line. An horizontal line (slope 0) indicates no resolution, whereas a 45-degree line indicates perfect resolution.

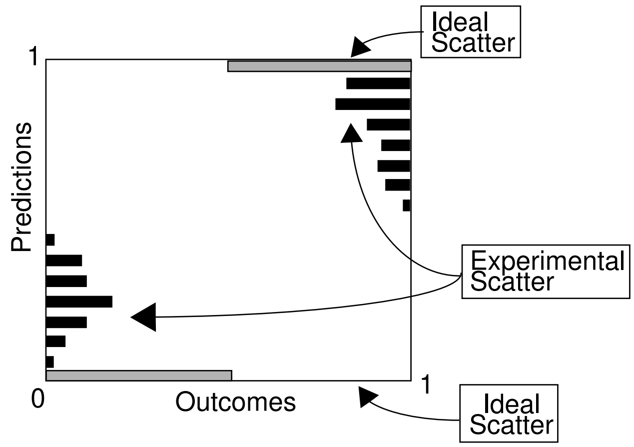
Scatter, or variability, is a measure of the “noisiness” in the data, and is represented by horizontal bars composing rotated histograms at each side of the covariance graph. Scatter is calculated as

$$scatter = [N_1\ Var(f_1) + N_0\ Var(f_0)] / (N_1 + N_0)$$

where N_1 is the number of patients with outcome “1,” N_0 is the number of patients with outcome “0,” f_1 is the average prediction for patients with outcome “1,” f_0 is the average prediction for patients with outcome “0,” and $Var(f_1)$ is the variance of f.

A perfect scatter would be zero, as shown in Figure 6.7. A large scatter means that patients with a given outcome have predictions that have high variance (e.g., predictions that vary from 0.5 to 1 for patients with real outcome “1”), which is often obtained when noisy data is introduced. Note that is it possible for the predictions to have large scatter (an undesirable feature), but still have high resolution. For example, if all patients with outcome “1” have predictions that are uniformly distributed in the interval of 0.5 to 1 and all patients with outcome “0” have predictions that are uniformly distributed in the interval 0 to 0.499, a large scatter and a large area under the ROC curve will result.

Figure 6.7. Assessing scatter in a covariance graph.

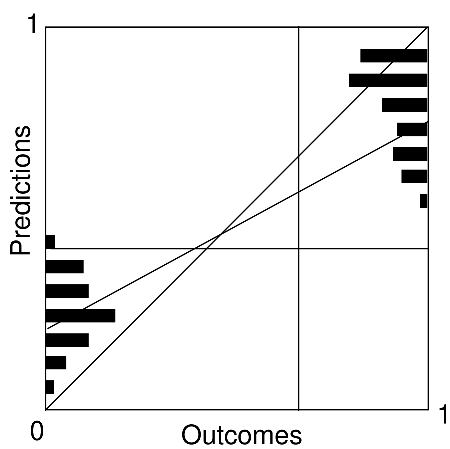


In a perfect scatter, all predictions for cases with outcome “0” would have predictions “0,” and cases with outcome “1” would have predictions of “1,” so there would be no variance.

Conversely, a small scatter just means that the predictions have low variance (which is desirable), but gives no indication as to whether these predictions are well calibrated or discriminatory. For example, a zero scatter is obtained by making the same prediction for all patients.

Figure 6.8 shows a complete covariance graph.

Figure 6.8. Complete covariance graph.



The complete covariance graph shows calibration, resolution, and scatter.

6.6 Methods Used in This Work

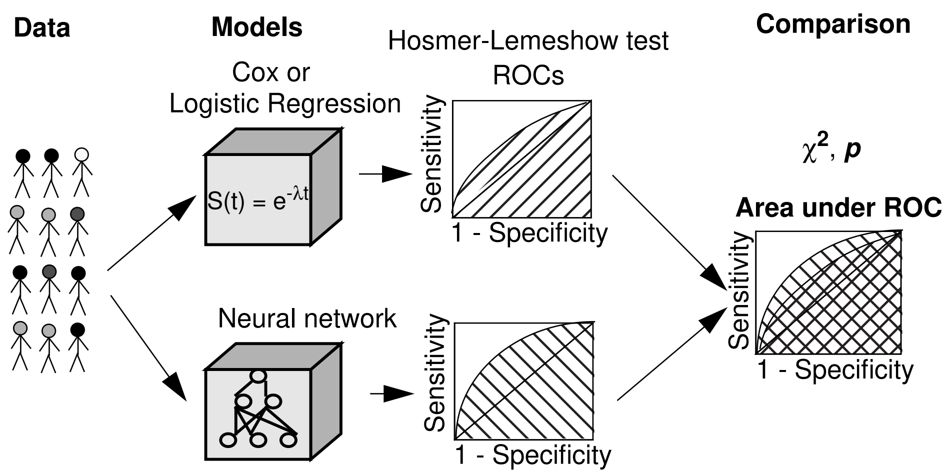
Calibration and resolution are complementary, and should be assessed for every model. As discussed previously, certain models can provide predictions that result in high calibration but poor resolution and other models can be highly discriminatory but poorly calibrated. Depending on the use of the model, calibration may have precedence over resolution, or vice versa. In the case of most diagnostic tests, for example, the most discriminatory threshold is chosen and a positive or negative interpretation is produced for a given patient. The test does not provide a probability that a patient has a disease, but a value that, if higher or lower than the threshold, will determine a positive or negative result. In other cases, such as providing prognostic estimates for a specific patient, calibration may take precedence over resolution. The physician wants to provide an estimate that is closer to the average outcome for a group of patients with similar characteristics. If, however, the goal is to discriminate patients with good and bad prognoses to allocate resources accordingly, resolution may take precedence. In the medical literature, calibration and resolution are frequently used in isolation. The use of both measures to evaluate a study facilitates the interpretation of results and should be done more often.

It is neither feasible nor necessary to apply all evaluation methods described in the chapter to the experiments of this work. Although certain measures of calibration and resolution help to determine whether gross differences between two models exist, and graphical methods help us to visualize these gross differences, but we are often confronted with models that are not grossly different. Furthermore, for several evaluation methods described here, there is no way to calculate standard errors and test statistical significance. Given that the Hosmer-Lemeshow test and the Wilcoxon statistic provide a means to statistically test the null hypothesis that any two given models have the same classification performance in terms of calibration and resolution, respectively, and that both these features are important in evaluating the performance of a classification model, they were chosen as the primary evaluation methods for this work.

All evaluation methods described in this chapter fail to take into account asymmetric loss functions (the penalty for false positives is the same as that for false negatives). These classical measures of predictive accuracy can be highly misleading depending on the circumstances in which they are used. If predictions for model A result in a larger area under the ROC curve than that resulting from predictions for model B using the same set of cases, it does not follow necessarily that model A is always better than model B. It may be the case that the area under the ROC curve for model A within a certain region of interest is in fact smaller than that of model B. In this case, model B would be considered the best. The results of this work describe the comparison of different models for predictions. The best way to use these predictions for decision making requires a decision-analytic approach that includes. among other things, the assessment of utilities for every outcome and the calculation of expected values, which are beyond the scope of this work.

Figure 6.9 shows the stages of the evaluation of resolution for different models.

Figure 6.9. Stages of evaluation.



The test sets are processed by the two models. The Hosmer-Lemeshow test is applied to the results of each model, and χ^2 and p are calculated for a given significance level α. The highest p defines the model with best calibration. Sensitivities and specificities for each model at each cut-off are made, and ROCs are built. Comparison of the areas under the ROCs defines the best model in terms of resolution.

CHAPTER 7

Models of Survival using the Framingham Data Set

This chapter describes the experiments that were conducted, using the Framingham data set, to test the hypotheses that (a) sequential models produce better prognostic indices than nonsequential models, and (b) neural network models produce better prognostic indices than logistic regression models. As we will see in the next sections, both the sequential neural network and the sequential logistic regression models performed better than their standard counterparts. The performances of neural network and logistic regression models were similar.

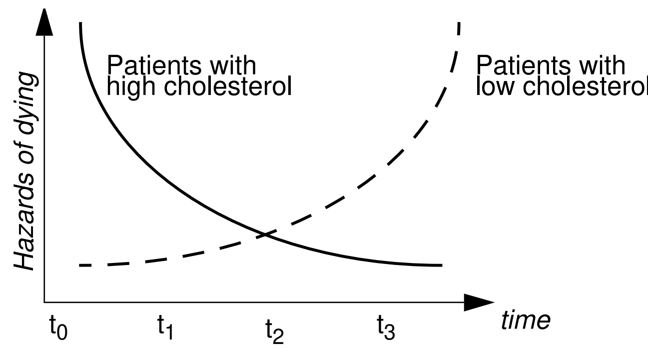
Section 7.1 introduces the domain of this study, describing the importance of modeling prognosis of coronary heart disease (CHD) development, listing currently recognized risk factors, and explaining the current models for prediction of CHD. Section 7.2 describes the logistic regression and the neural network models used in this work, emphasizing the experimental design. Section 7.3 compares neural network and logistic regression models. Section 7.4 compares standard and sequential methods. Section 8.5 discusses the results of this study.

7.1 Prognosis of CHD Development and the Framingham Data Set

CHD is the leading cause of death in industrialized countries. Although the death rate from CHD has decreased in recent years, as a result of both changes in lifestyle and therapeutic interventions, acute myocardial infarction (a manifestation of CHD) is still a major cause of death and disability in the United States. Major identified risk factors for CHD include (1) elevated blood pressure [Kannel, 1993 and 1990], (2) elevated serum cholesterol [Kreger, 1994; Kannel, 1993; Castelli, 1992; Anderson, 1991; Wong, 1991], (3) elevated hematocrit [Gagnon, 1994], (4) elevated serum glucose [Kannel, 1990], (5) obesity [Higgins, 1993; Posner 1993; Kannel, 1991], (6) increased serum fibrinogen [Kannel, 1992 and 1990], (7) elevated heart rate [Gillman, 1993], (8) advanced age [Jenner, 1993], (9) male gender [Jenner, 1993], and (10) cigarette smoking [Freund, 1993; Kannel, 1990]. Kannel [1993] has pointed out that *"no single factor has been found to be essential or sufficient in the evolution of the disease,"* emphasizing the importance of multivariate analyses to predict the development or aggravation of CHD. Thus, the risk associated with any one of these factors should not be calculated independent of the other factors [Percy, 1993]. For example, one study has shown that the risk associated with elevated levels of cholesterol varies according to age: while increased levels are harmful for middle-aged males, they seem to be associated with *no increase* in risk for CHD mortality (or even a *decrease* in overall mortality) for the elderly [Kronmal, 1993]. For some groups, it still unclear whether cholesterol-reduction measures (especially those based on medications) are associated with an increase in survival. Some authors recommend lowering cholesterol levels (including the use of lipid-lowering medications) for elderly patients with high cholesterol levels, whereas other are more cautious [Drown, 1994; Temple 1994; Pacala, 1994; Capurso, 1992].

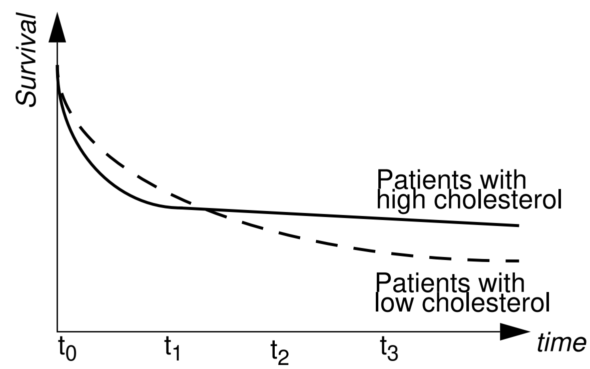
Different models can be used to study CHD development, given a set of data. The Cox proportional hazards model is one of them. However, the form of the function that relates risk (or hazards) of death from CHD to time for two different cholesterol-based strata may be of the type shown in Figure 8.1, indicating that modeling of disease progression by Cox proportional hazards models may not be appropriate, since hazards may not remain proportional over time for these strata. That is, a survival curve for patients with high cholesterol levels may cross that of patients with low cholesterol levels after a certain time, as in the example shown in Figure 8.2. This is clearly a violation of the proportional hazards assumption.

Figure 7.1. Hypothetical example of nonproportional hazards.



The hazards for patients with high cholesterol may not always be proportionally higher than those for patients with low cholesterol.

Figure 7.2. Survival curves crossing for the hypothetical example.

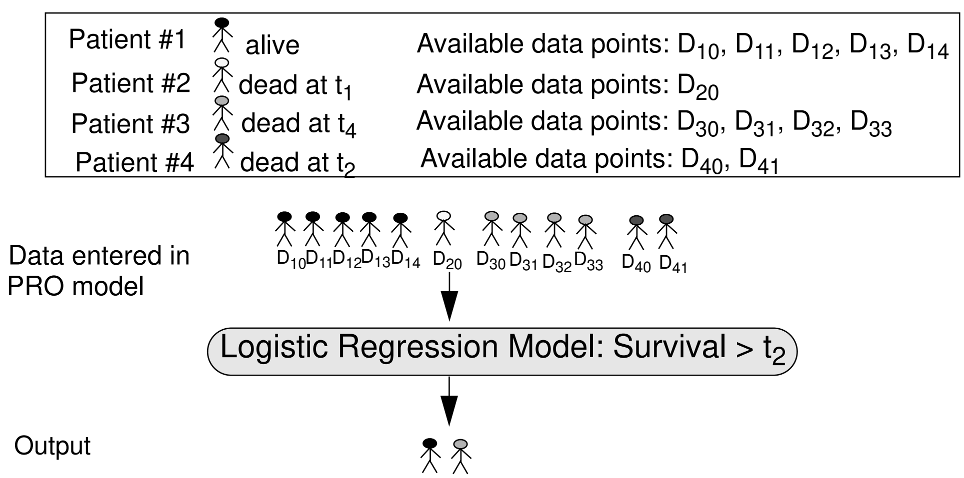


If the hazards are not proportional, it is likely that survival curves for patients with high and low cholesterol will cross after a certain time, which violates the proportionality assumption.

The Framingham Heart Study provided a fundamental source of data concerning the development of CHD for several researchers [Dawber, 1980]. It began in 1948 with a cohort of 5209 men and women aged 30 to 62 who did not have cardiac disease. Every two years all patients were assessed for cardiovascular and oncologic diseases, as well as mortality. Fewer than 2 percent of the participants have been lost to follow-up [Posner, 1993]. Information was collected at baseline concerning lifestyle (diet, physical activity, use of cigarettes, alcohol, and coffee), atherogenic traits (blood lipids, blood pressure, blood glucose, and fibrinogen), and ECG [Kannel, 1990]. CHD events included angina pectoris, unstable angina, and myocardial infarction [Ho, 1993]. The causes and date of death have been recorded for all cases. Traditionally, the Framingham data have been analyzed by logistic regression models [D'Agostino, 1990, Cupples, 1988]. In their pure form, logistic regression models do not entail any assumption of hazards proportionality.

The associations between the covariates (risk factors) and CHD have been modeled mainly by pooled-logistic regression analysis. Figure 7.3 shows a kind of pooled-logistic regression used [Cupples, 1988], the Pool of Repeated Observations (PRO), in which repeated observations are pooled in intervals and presented to the logistic regression model. In pooled-logistic regression models, data from the same individual may be entered several times (e.g., once every two years).

Figure 7.3. Pool of Repeated Observations (PRO).



In the PRO model, data from the same patient may be entered more than once if the patient has survived for a long period. No information is lost, but there is an implicit assumption that repeated measurements of the same patient are independent of each other.

The PRO model can use censored data and can update the risk factors at the beginning of each observation interval, allowing for time-dependent covariates. D'Agostino [1990] has shown that the PRO and the Cox models are asymptotically equivalent when the intervals between measurements are short, and the probability of an event within the interval is small. Ingram [1989] has shown that if the proportion of deaths is not high (<20 percent), the cumulative and person-time logistic models produce parameter estimates that are similar to those arrived at using the Cox model. Ingram also states that mild violations of the proportional hazards assumption do not significantly influence the parameter estimates from the Cox or logistic models.

PRO usually assumes that only the current risk profile is necessary to predict the event (Markovian assumption). Necessary assumptions for applying this model include the absence of secular trends and the same underlying risk of disease in each interval [Cupples, 1988]. The risk related to a certain combination of variable values is obtained for a given interval and is multiplied to obtain the risk in multiples of that interval. For example, if a risk of developing CHD in two years is 0.02 for a patient with a given set of variable values, the risk of this patient developing CHD in four years is 2*0.02=0.04. Implicit in this model is a proportionality assumption. It is easy to see that if the risk is high or the interval being considered is long, a risk of 1 or higher will be obtained by this method, which would lead to the false result that virtually every patient would develop CHD after a certain time.

The majority of the models described in the literature that used either Cox proportional hazards or logistic regression to study CHD had the goal of defining important variables in the development of the disease. These models were not designed to provide a prognosis for patients with a given set of variable values and therefore were not evaluated using a different set of cases. The measures of goodness-of-fit provide a clue to the generalization ability of these models, but no cross-validation or resampling methods—such as the ones described in Chapter 2— have been utilized. Overfitting may have occurred. Furthermore, these models may be inadequate for long-term prognoses.

Since the objective of this study was to compare the accuracy of the prognostic indices produced by two different methods (standard, or nonsequential, and sequential) and two different types of models (logistic regression and neural networks) on a previously unseen set of cases, and for short and long-term prognosis, I could not use the existing results from the literature as the "control" set, so I developed new logistic and neural network models.

7.2 Experimental Design and Results

The subset of the Framingham data set used in this study was obtained from Garber [1994], and described of 2594 men for whom data on cigarette smoking, total cholesterol, systolic and diastolic blood pressure, metropolitan relative weight, age, myocardial infarction and other CHD diagnoses, glucose intolerance, hematocrit, vital capacity, left ventricular hypertrophy diagnosis, and cause of death (when applicable) were available. The data set consisted of one entry for each patient's biannual examination. All patients were given an identification number in ascending order according to their appearance in the data set. Every third patient was assigned to a test set and all his exams were removed from the original data set, which became the training set. There was no imputation of values on the data. Training and test tests were made for each interval of two years of follow-up, consisting of 2/3 and 1/3 of the cases available for that interval, respectively. The number of entries (exams) and the distribution of CHD according to year of follow-up for training and test sets are show in Table 7.1. The balance of cases in each set is defined as

balance = min(CHD/Total, non-CHD/Total)

A balance of 0.5 means that the proportion of cases is 1:1, and is the one that best facilitates classification by statistical and neural network models.

Table 7.1. **Distribution of cases in training and test sets according to year of follow-up.**

	Training set				Test set			
Year of follow-up	**CHD**	**non-CHD**	**Total**	**Balance**	**CHD**	**non-CHD**	**Total**	**Balance**
2	189	5017	5206	0.0363	95	2496	2591	0.0367
4	344	4249	4593	0.0749	184	2116	2300	0.0800
6	499	3990	4489	0.1112	250	1987	2237	0.1118
8	648	3699	4347	0.1491	326	1846	2172	0.1501
10	794	3426	4220	0.1882	392	1703	2095	0.1871
12	946	2673	3619	0.2614	467	1367	1834	0.2546
14	1069	2013	3082	0.3469	531	1009	1540	0.3448
16	1162	1448	2610	0.4452	579	734	1313	0.4410
18	1236	941	2177	0.4322	611	495	1106	0.4476
20	1276	810	2086	0.3882	631	438	1069	0.4097
22	1315	350	1665	0.2102	655	194	849	0.2285

The last year with available follow-up was year 22.

Neural network and logistic regression models were built to test the hypotheses that (a) a sequential model could predict survival more accurately than a nonsequential model, and (b) a neural network model could predict survival more accurately than a logistic regression model that used the same covariates. Accuracy was determined by comparing predictions on a test set, according to measurements of calibration and resolution, as discussed previously in Chapter 6.

Independent variables are shown in Table 7.1. The same interactions among variables used by Garber [1994] were utilized in our experiment. The dependent variable was development of CHD.

Table 7.2. **Independent variables.**

name	description
AGE	age
HBP	hypertension
FVC	functional lung vital capacity
LVH	left ventricular hypertrophy
VEN	ECG's ventricular rate
SCL	serum cholesterol
SCLL1	serum cholesterol at first exam
AVEDP	average diastolic blood pressure
AVESP	average systolic blood pressure
MRW	metropolitan relative weight
CSM	number cigarettes smoked per day
GLI	glucose intolerance
AVESCL	average cholesterol
AGESPF	AGE*systolic blood pressure
AGEDPF	AGE*diastolic blood pressure
AGECSM	AGE*CSM
AGEGLI	AGE*GLI
AGEMRW	AGE*MRW
SCLSQ	SCL^2
AGESCL	AGE*SCL
AVESCLSQ	$AVESCL^2$
AGEAVDP	AGE*AVEDP
AGEAVSP	AGE*AVESP
AGAVSC	AGE*AVESCL
AGAVSCSQ	$AGE*AVESCLS^2$
AGESCLSQ	$AGE*SCL^2$
MRWCSM	MRW*CSM

Standard Logistic Regression Model. A standard logistic regression model, similar to the one shown in Figure 7.4, was built for each biannual interval with the variables presented earlier in this chapter.

Figure 7.4. Standard logistic regression model: Framingham data.

$$\mathrm{Prob}(\mathrm{CHDh}_{t\zeta}) = \frac{e^{(a*age_{t\zeta}+b*gender+c*bp_{t\zeta}+d*chol_{t\zeta}+e*smoking_{t\zeta}+f*weight_{t\zeta})}}{1+e^{(a*age_{t\zeta}+b*gende_{t\zeta}r+c*bp_{t\zeta}+d*chol_{t\zeta}+e*smoking_{t\zeta}+f*weight_{t\zeta})}}$$

In a standard logistic regression model, the probability of an event at time t_ζ can be calculated as a function of the available variables. Maximum likelihood estimates (coefficients a,b,c, etc.) are produced.

The SAS/STAT procedure LOGISTIC [SAS, 1990], using the IRLS algorithm, was used in the training set to calculate the maximum likelihood estimators of the regression parameters. No variable selection procedure was used. In all models, the *-2 log L* statistics[1] provided a χ^2 value that corresponded to $p<0.05$, indicating that the coefficients of the explanatory variables were different from zero. The variables with the largest standardized coefficients were SCL, SCL1, and AVESCL (see Table 7.1 for description), confirming the importance of total cholesterol values in the development of CHD.

Table 7.1 shows the calibration of the standard logistic regression model for training and test sets.

Table 7.3. **Calibration of standard logistic regression models.**

	Test set	
Year of follow-up	χ^2	***p***
2	15.5092	0.04997
4	16.0343	0.04189
6	18.4838	0.01788
8	20.3515	0.00909
10	22.6141	0.00390
12	17.0659	0.02943
14	9.7718	0.28141
16	5.0194	0.75551
18	12.2332	0.14110
20	7.2723	0.49704
22	17.8476	0.02240

[1] *L* is the value ot the likelihood function when the parameters are replaced by their maximum likelihood extimates [Collet, 1994].

Four of the 11 models in the test set were well calibrated (p>0.05).

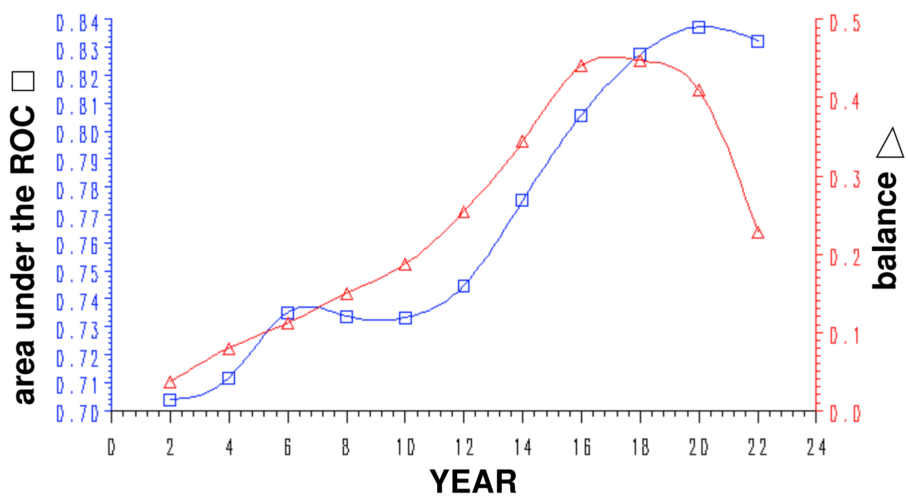
Table 7.1 shows the areas under the ROC curves for the test sets.

Table 7.4. **Resolution of standard logistic regression models.**

	Test set	
Year of follow-up	**area under the ROC curve**	***standard error***
2	0.6717	0.0261
4	0.7277	0.0186
6	0.7356	0.0154
8	0.7337	0.0137
10	0.7332	0.0127
12	0.7482	0.0123
14	0.7791	0.0120
16	0.8126	0.0116
18	0.8400	0.0116
20	0.8432	0.0118
22	0.8436	0.0153

The best discrimination was achieved for the last interval, or follow-up of 22 years. Figure 7.5 shows how resolution is related to data balance.

Figure 7.5. Resolution and data balance in standard logistic regression model.

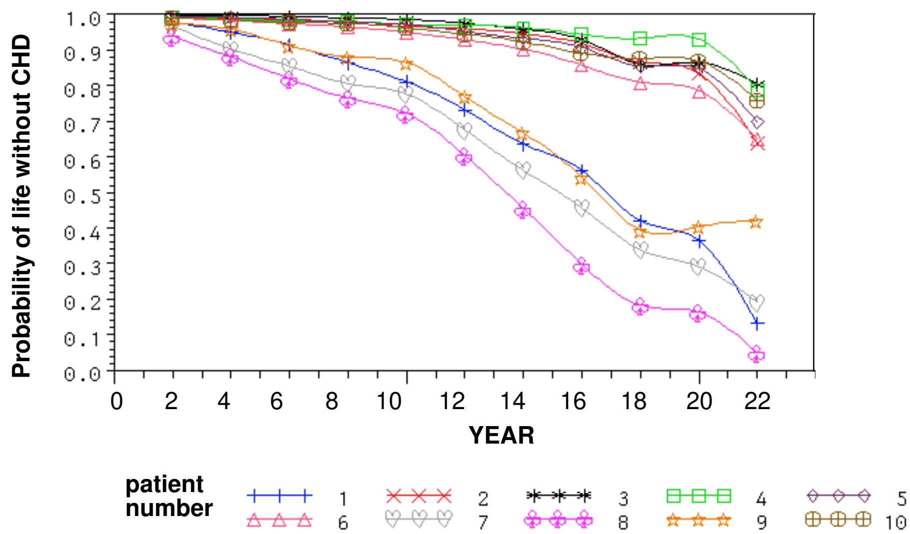


Resolution is represented by squares and scaled in the left axis. Balance is represented by triangles and scaled in the right axis.

The predictions had low resolution for years in which data were highly unbalanced (e.g., years 2 and 4). Resolution increased for years in which data were balanced.

Every two years, a different standard logistic regression model was used to assess the probability of CHD development for all patients. Predictions for 11 intervals where plotted for 10 patients, producing the survival curves in Figure 7.6.

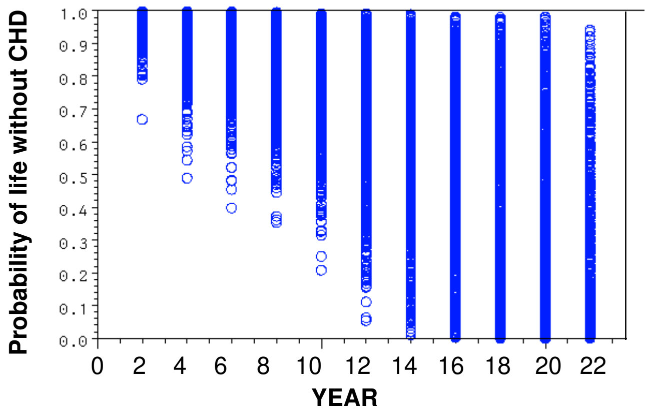
Figure 7.6. Survival curves for ten patients using standard logistic regression.



Since the models have no relation to each other, survival curves that are not monotonically decreasing, although impossible in theory, can be produced, such as for patient number 9.

The range of probabilities produced by the model in the first intervals was limited (e.g., the minimum probability of survival in Year 2 was 0.67, and most predictions would correspond to values over 0.8), as shown in Figure 7.7. The range of probabilities in the last intervals was wider, indicating a higher balance in those intervals.

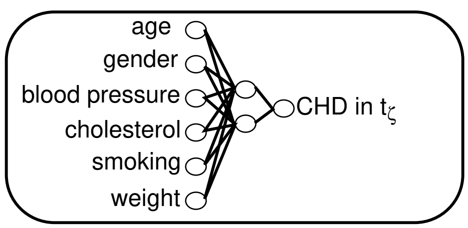
Figure 7.7. Range of probabilities for standard logistic regression model.



Predictions for the first intervals tended to be conservative, avoiding small values. Predictions for the last intervals were better distributed, indicating a better data balance.

Standard Neural Network Model. A standard neural network model, similar to the one shown in Figure 7.8, was built for each biannual interval with the variables presented earlier in this chapter.

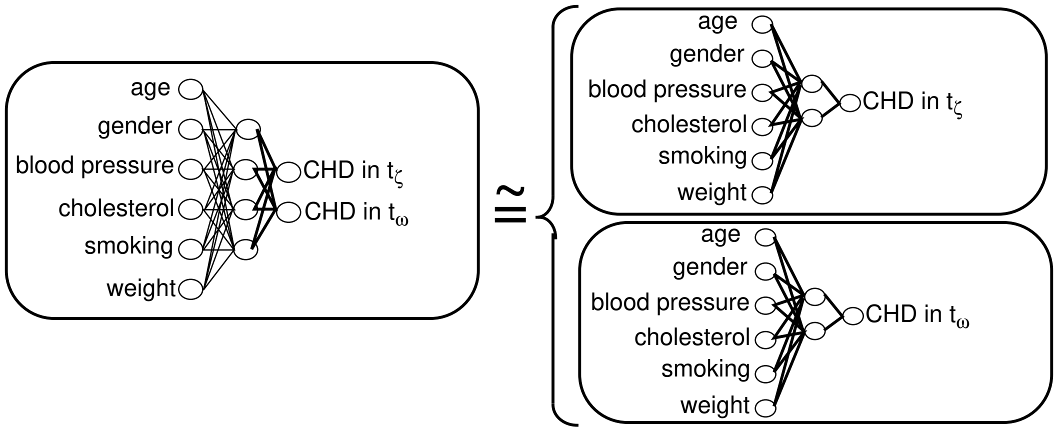
Figure 7.8. Standard neural network model: Framingham data.



In a standard neural network model, survival is predicted for one interval at a time, and no information on previous intervals is provided.

The models constructed for each interval could have been combined in an overall standard neural network model, such as the one shown in Figure 7.9, but that was not done since an overall standard neural network model could not handle censored data (outcomes for all intervals would be necessary) and no equivalent logistic regression model could be built for comparison.

Figure 7.9. Equivalence of standard neural network models: Framingham data.



The model on the right was the one chosen for this experiment. The model on the left is more economical (fewer weights) than the one at the right, but the nonsequential nature of the processing of information is the same. Furthermore, it is advantageous to use the model on the right, since it can use censored data and it has a logistic regression equivalent.

Neural networks were constructed with ten hidden nodes and trained by minimization of cross-entropy error using the quickpropagation algorithm (an optimization of backpropagation) developed by Fahlman [1988]. Overfitting was monitored by the error in half of the training set. The software NevProp2, developed by Goodman and colleagues [1994] was used.

Table 7.1 shows the calibration of the model for the test sets.

Table 7.5. **Calibration of standard neural network models.**

	Test set	
Year of follow-up	χ^2	***p***
2	15.0118	0.0589
4	11.1389	0.1939
6	19.6175	0.0118
8	30.3247	0.0001
10	23.6363	0.0026
12	11.6443	0.1677
14	9.3273	0.3154
16	6.7588	0.5628
18	26.1660	0.0009
20	22.3739	0.0042
22	12.7683	0.1200

Six of the 11 models in the test set had good calibration ($p>0.05$).

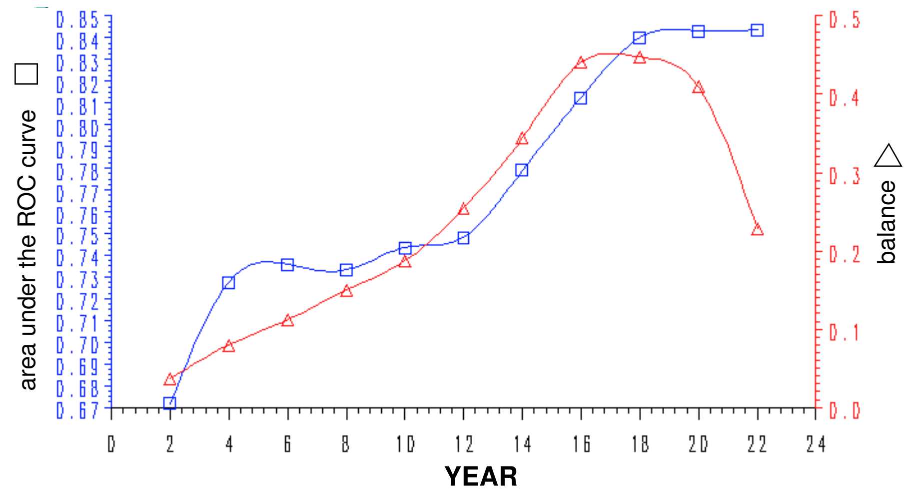
Table 7.1 shows the areas under the ROC curves for the test sets.

Table 7.6. **Resolution of standard neural network models.**

	Test set	
Year of follow-up	**area under the ROC curve**	***standard error***
2	0.7038	0.0242
4	0.7117	0.0190
6	0.7352	0.0152
8	0.7337	0.0138
10	0.7333	0.0130
12	0.7448	0.0123
14	0.7752	0.0121
16	0.8059	0.0119
18	0.8275	0.0122
20	0.8374	0.0122
22	0.8324	0.0163

The best discrimination was achieved for a follow-up of 20 years. Figure 7.10 shows how resolution is related to data balance.

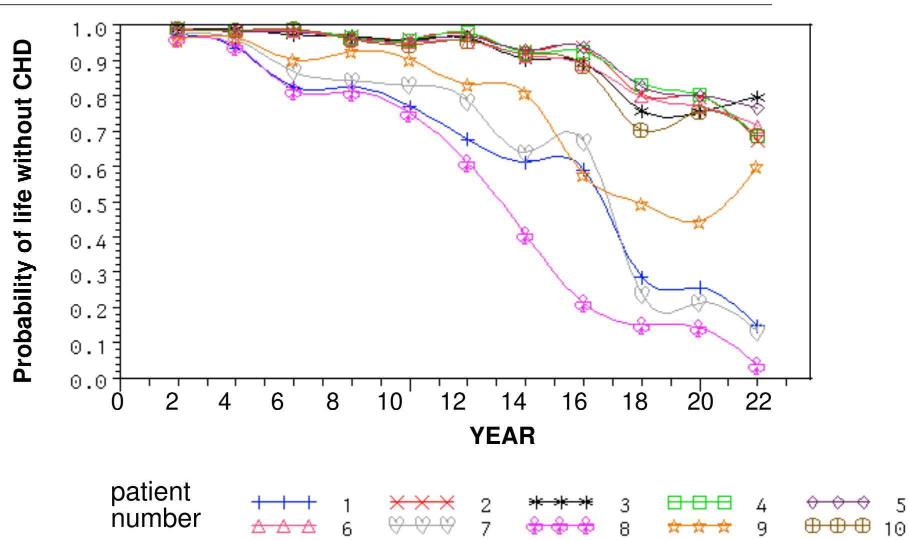
Figure 7.10. Resolution and data balance in standard neural network model.



Resolution is represented by squares and scaled in the left axis. Balance is represented by triangles and scaled in the right axis.

Figure 7.11 shows an example of ten survival curves produced by the standard neural network models.

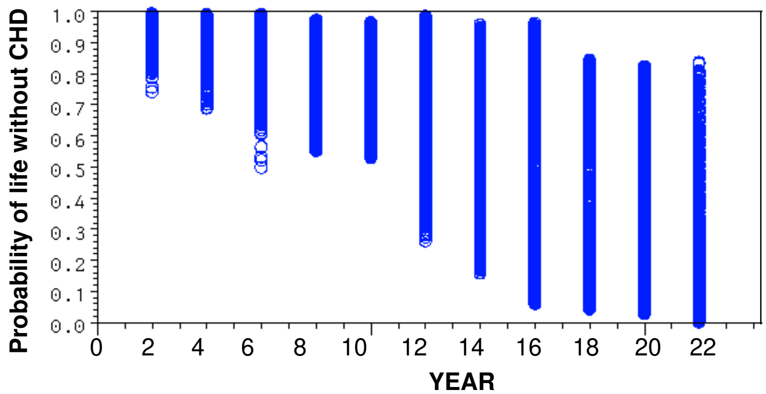
Figure 7.11. Survival curves for ten patients using standard neural networks.



Since the models have no relation to each other, survival curves that are not monotonically decreasing, although impossible in theory, can be produced, such as for patient number 9.

The range of probabilities produced by the model in the first intervals was limited (e.g., the minimum probability of survival in Year 2 was 0.74, and most predictions would correspond to values over 0.8), as shown in Figure 7.12. The range of probabilities in the last intervals was wider, indicating a higher balance of data in those intervals.

Figure 7.12. Range of probabilities for standard neural network model for all patients.



Predictions for the first intervals tended to be conservative, avoiding small values. Predictions for the last intervals were better distributed, indicating a better data balance

Sequential Logistic Regression Model. A sequential model, similar to the one shown in Figure 7.13, was built for each possible pair of intervals, resulting in 110 sequential models (e.g., a sequential model in which the output of the standard logistic regression model for Year 2 was entered as input to a sequential model for Year 4).

Figure 7.13. Sequential logistic regression model.

$$\text{Prob}(\text{CHD}_{t\zeta}) = \frac{e^{(a^*\text{age}_{t\zeta}+b^*\text{gender}+c^*\text{bp}_{t\zeta}+d^*\text{chol}_{t\zeta}+e^*\text{smoking}_{t\zeta}+f^*\text{weight}_{t\zeta})}}{1+e^{(a^*\text{age}_{t\zeta}+b^*\text{gende}_{t\zeta}\text{r}+c^*\text{bp}_{t\zeta}+d^*\text{chol}_{t\zeta}+e^*\text{smoking}_{t\zeta}+f^*\text{weight}_{t\zeta})}}$$

$$\text{Prob}(\text{CHD}_{t\omega}) = \frac{e^{(g^*\text{Prob}(\text{CHDh}_{t\zeta})+\ a^*\text{age}_{t\omega}+b^*\text{gender}+c^*\text{bp}_{t\omega}+d^*\text{chol}_{t\omega}+e^*\text{smoking}_{t\omega}+f^*\text{weight}_{t\omega})}}{1+e^{(\text{Prob}(\text{CHDh}_{t\zeta})+a^*\text{age}_{t\omega}+b^*\text{gender}_{t\omega}+c^*\text{bp}_{t\omega}+d^*\text{chol}_{t\omega}+e^*\text{smoking}_{t\omega}+f^*\text{weight}_{t\omega})}}$$

In a sequential logistic regression model, the probability of an event at time t_ζ obtained from the standard model can be entered as input to the sequential logistic regression model that predicts the event at time $t\omega$.

The SAS/STAT procedure LOGISTIC [SAS, 1990], using the IRLS algorithm, was used in the training set to calculate the maximum likelihood estimators of the regression parameters. No variable selection procedure was used.

In all models, the *-2 log L* statistics provided a χ^2 value that corresponded to $p<0.05$, indicating that the coefficients of the explanatory variables were different from zero. The magnitude of the standardized coefficient of a variable can be interpreted as the importance of that variable to the model. The higher the standardized coefficient, the greater the contribution of that variable to the model. In the sequential logistic regression model, the variables with the largest standardized coefficients were again the ones corresponding to the prediction of CHD development in another interval, SCL, SCL1, and AVESCL, confirming the importance of total cholesterol values in the development of CHD. The independent variable that corresponded to a prediction in a certain year that was provided for the sequential equation always had high standardized coefficients in the final sequential equations, often among the five higher standardized coefficients, meaning that providing information on a certain year was indeed taken into account to construct the model.

Of the 110 sequential logistic regression models, 79 were well calibrated ($p>0.05$) using the Hosmer-Lemeshow test, as shown in Table 7.7.

.

Table 7.7. Calibration of sequential logistic regression models.

	Year											
	2		4		6		8		10		12	
Informative Year	χ^2	p	χ^2	p	χ^2	p	χ^2	p	χ^2	p	χ^2	p
2			16.2622	0.04	16.4773	0.04	16.9170	0.03	18.8832	0.02	16.9412	0.03
4	9.2603	0.32			14.6006	0.07	19.5386	0.01	21.2465	0.01	15.9692	0.04
6	9.8167	0.28	13.3485	0.10			22.7839	0.00	18.9354	0.02	12.7160	0.12
8	9.0782	0.34	9.2630	0.32	25.6066	0.00			18.3071	0.02	13.3926	0.10
10	6.4889	0.59	11.5179	0.17	23.0568	0.00	14.8270	0.06			12.4154	0.13
12	8.8974	0.35	10.8185	0.21	8.9106	0.35	6.3869	0.60	6.8580	0.55		
14	7.6828	0.47	13.0043	0.11	8.4339	0.39	14.7627	0.06	17.8473	0.02	10.8511	0.21
16	11.7466	0.16	11.0953	0.20	16.0566	0.04	12.7058	0.12	25.7655	0.00	13.5346	0.09
18	9.8240	0.28	9.0164	0.34	13.9674	0.08	7.0831	0.53	22.4416	0.00	11.8906	0.16
20	12.5060	0.13	8.2823	0.41	12.7024	0.12	7.4784	0.49	9.1290	0.33	14.3677	0.07
22	12.6895	0.12	14.3858	0.07	14.3320	0.07	14.9852	0.06	16.4859	0.04	10.4696	0.23

	14		16		18		20		22	
Informative Year	χ^2	p	χ^2	p	χ^2	p	χ^2	p	χ^2	p
2	10.6408	0.22	4.3869	0.82	9.9329	0.27	6.0103	0.65	18.9968	0.01
4	10.5416	0.23	4.2342	0.84	9.2062	0.33	6.7098	0.57	20.8384	0.01
6	12.9967	0.11	8.0011	0.43	9.9158	0.27	6.0370	0.64	20.2341	0.01
8	14.7925	0.06	6.5203	0.59	7.9279	0.44	6.2640	0.62	20.9151	0.01
10	16.1173	0.04	5.4484	0.71	8.2986	0.40	7.4760	0.49	22.0034	0.00
12	10.7715	0.21	7.8378	0.45	3.9319	0.86	7.6052	0.47	22.6384	0.00
14			8.3419	0.40	5.8397	0.67	6.7003	0.57	18.5448	0.02
16	6.8376	0.55			7.7899	0.45	8.0565	0.43	17.1252	0.03
18	13.6960	0.09	7.8347	0.45			7.4139	0.49	18.9089	0.02
20	15.1968	0.06	10.7171	0.22	6.6289	0.58			23.9396	0.00
22	8.9876	0.34	7.8237	0.45	25.9936	0.00	21.7507	0.01		

Table 7.8 shows the areas under the ROC curves and standard errors for all sequential logistic regression models.

Table 7.8. **Resolution of sequential logistic regression models.**

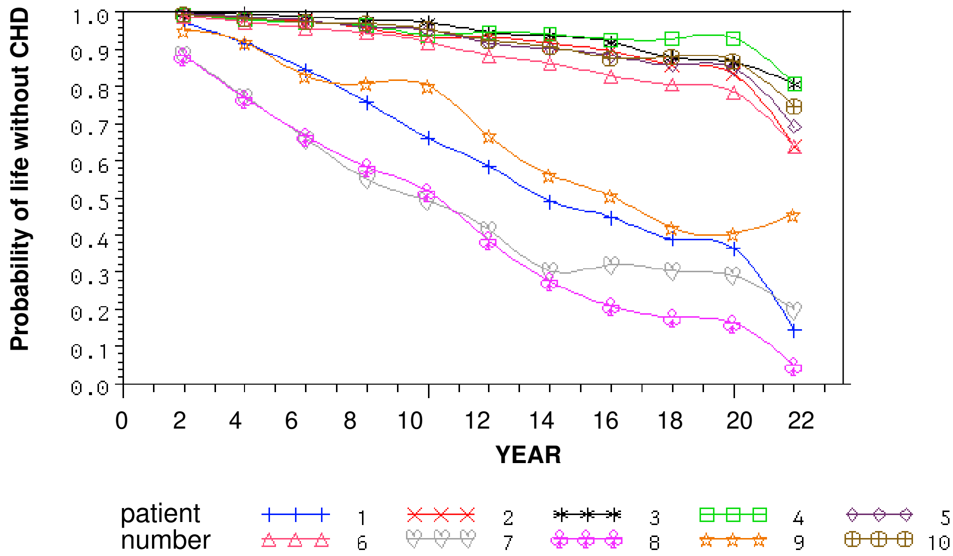
Informative Year	Year 2 *W*	2 *s.e.*	4 *W*	4 *s.e.*	6 *W*	6 *s.e.*	8 *W*	8 *s.e.*	10 *W*	10 *s.e.*
2			0.7131	0.0257	0.7175	0.0253	0.7310	0.0245	0.7347	0.0246
4	0.7276	0.0187			0.7323	0.0184	0.7429	0.0182	0.7472	0.0184
6	0.7333	0.0158	0.7351	0.0156			0.7473	0.0154	0.7536	0.0154
8	0.7326	0.0138	0.7335	0.0137	0.7346	0.0138			0.7424	0.0137
10	0.7436	0.0127	0.7438	0.0127	0.7443	0.0127	0.7457	0.0127		
12	0.7485	0.0124	0.7488	0.0123	0.7484	0.0124	0.7489	0.0123	0.7488	0.0123
14	0.7789	0.0120	0.7792	0.0120	0.7798	0.0120	0.7797	0.0120	0.7798	0.0120
16	0.8129	0.0117	0.8127	0.0117	0.8130	0.0116	0.8133	0.0116	0.8134	0.0116
18	0.8399	0.0117	0.8399	0.0117	0.8398	0.0117	0.8399	0.0117	0.8394	0.0117
20	0.8429	0.0119	0.8431	0.0119	0.8432	0.0119	0.8431	0.0119	0.8430	0.0119
22	0.8433	0.0154	0.8433	0.0154	0.8439	0.0154	0.8434	0.0155	0.8423	0.0156

Informative Year	12 *W*	12 *s.e.*	14 *W*	14 *s.e.*	16 *W*	16 *s.e.*	18 *W*	18 *s.e.*	20 *W*	20 *s.e.*
2	0.7328	0.0255	0.7507	0.0252	0.7518	0.0254	0.7480	0.0249	0.7442	0.0250
4	0.7534	0.0185	0.7691	0.0184	0.7742	0.0185	0.7753	0.0183	0.7718	0.0184
6	0.7583	0.0157	0.7759	0.0155	0.7875	0.0153	0.7881	0.0157	0.7884	0.0156
8	0.7516	0.0139	0.7719	0.0138	0.7857	0.0138	0.7896	0.0141	0.7899	0.0142
10	0.7568	0.0128	0.7819	0.0127	0.8001	0.0127	0.8064	0.0131	0.8088	0.0131
12			0.7803	0.0122	0.8060	0.0121	0.8195	0.0124	0.8227	0.0125
14	0.7795	0.0120			0.8098	0.0119	0.8307	0.0120	0.8360	0.0121
16	0.8127	0.0117	0.8132	0.0116			0.8409	0.0116	0.8478	0.0116
18	0.8391	0.0117	0.8393	0.0117	0.8401	0.0117			0.8483	0.0116
20	0.8428	0.0119	0.8431	0.0119	0.8436	0.0119	0.8432	0.0119		
22	0.8422	0.0156	0.8425	0.0156	0.8424	0.0155	0.8436	0.0154	0.8432	0.0155

Informative Year	22 *W*	22 *s.e.*
2	0.7020	0.0266
4	0.7385	0.0200
6	0.7542	0.0175
8	0.7519	0.0165
10	0.7713	0.0158
12	0.7901	0.0156
14	0.8130	0.0153
16	0.8356	0.0146
18	0.8441	0.0147
20	0.8373	0.0157

Figure 7.14 shows an example of ten survival curves produced by the sequential logistic regression models in which predictions in Year 20 were provided.

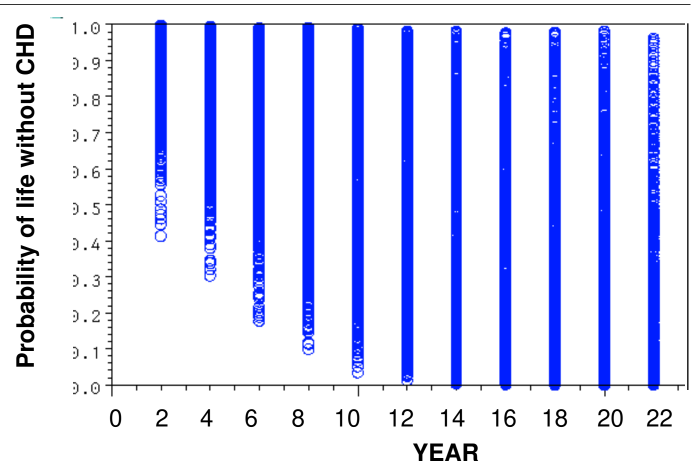
Figure 7.14. Survival curves using logistic regression and information of Year 20.



Nonmonotonic curves, such as the one corresponding to patient number 9, are still produced in sequential logistic regression models, but they are not as steep.

The range of probabilities is shown in Figure 7.15.

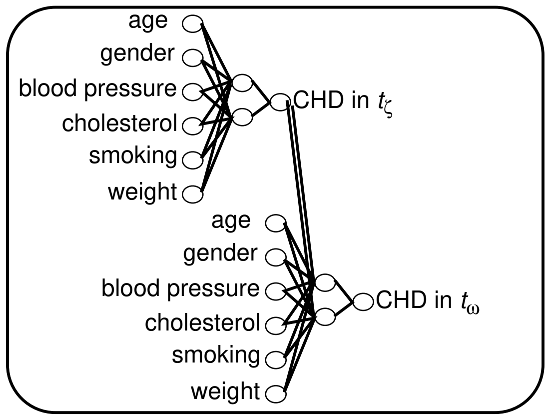
Figure 7.15. Range of probabilities for logistic regression using information from Year 20.



The range of probabilities for sequential models is wide, including the ones produced for the first intervals.

Sequential Neural Network Model. A sequential model, similar to the one shown in Figure 7.16, was built for each possible pair of intervals, resulting in 110 sequential models, just as in the sequential logistic regression model described above.

Figure 7.16. Sequential neural network model.



In a sequential neural network model, the probability of an event at time t_ζ obtained from the standard neural network model can be entered as input to the sequential neural network model that predicts the event at time t_ω.

Neural networks were constructed with ten hidden nodes and trained by minimization of cross-entropy error using the quickpropagation algorithm. Overfitting was monitored by the error in half of the training set. The software NevProp2 was used.

Of the 110 sequential neural networks, 48 were well calibrated (p>0.05) using the Hosmer-Lemeshow test, as shown in Table 7.9.

Table 7.9. **Calibration of sequential neural network models.**

		Year											
		2		4		6		8		10		12	
		χ^2	p	χ^2	p	χ^2	p	χ^2	p	χ^2	p	χ^2	p
Informative Year	2			13.1435	**0.11**	14.6240	**0.07**	8.5727	**0.38**	52.6269	0.00	18.3555	0.02
	4	9.8212	0.28			12.8448	0.12	14.9645	0.06	41.8917	0.00	22.6315	0.00
	6	3.3991	0.91	11.6007	0.17			25.1379	0.00	42.5970	0.00	29.1821	0.00
	8	17.0644	0.03	14.9963	0.06	10.6357	0.22			52.0179	0.00	12.2629	0.14
	10	18.2274	0.02	13.6959	0.09	30.4125	0.00	29.3929	0.00			7.9547	0.44
	12	12.9322	0.11	8.3186	0.40	43.6185	0.00	38.5990	0.00	35.2165	0.00		
	14	8.2831	0.41	13.2362	0.10	29.9100	0.00	19.6264	0.01	48.8063	0.00	10.2974	0.24
	16	6.1119	0.63	8.7042	0.37	26.7849	0.00	26.7072	0.00	26.7886	0.00	11.6152	0.17
	18	9.6610	0.29	8.9642	0.35	51.4274	0.00	71.1437	0.00	18.1969	0.02	36.4449	0.00
	20	10.2685	0.25	22.5018	0.00	27.5929	0.00	13.0709	0.11	40.1797	0.00	19.4396	0.01
	22	9.6340	0.29	22.3997	0.00	19.4347	0.01	55.7164	0.00	13.7669	0.09	13.5392	0.09

		14		16		18		20		22	
		χ^2	p	χ^2	p	χ^2	p	χ^2	p	χ^2	p
Informative Year	2	13.7267	0.09	5.5901	0.69	21.2171	0.01	23.1064	0.00	12.4202	0.13
	4	19.7395	0.01	5.7234	0.68	10.0610	0.26	21.2731	0.01	6.8863	0.55
	6	40.9677	0.00	9.4570	0.31	34.4733	0.00	26.6079	0.00	13.9140	0.08
	8	15.2342	0.05	3.0978	0.93	20.2522	0.01	19.9029	0.01	13.4868	0.10
	10	12.8500	0.12	10.3195	0.24	41.8349	0.00	39.8103	0.00	16.6063	0.03
	12	25.1583	0.00	7.6935	0.46	19.9963	0.01	51.1736	0.00	10.9975	0.20
	14			4.6081	0.80	23.5968	0.00	33.0921	0.00	13.3280	0.10
	16	15.0717	0.06			14.6768	0.07	23.9607	0.00	13.9382	0.08
	18	23.8345	0.00	13.6680	0.09			13.6152	0.09	10.3478	0.24
	20	14.7418	0.06	21.2347	0.01	21.8059	0.01			23.6344	0.00
	22	10.3695	0.24	7.0661	0.53	20.0486	0.01	11.6077	0.17		

Table 7.10 shows the areas under the ROC curves for the sequential neural network models.

Table 7.10. **Resolution of sequential neural network models.**

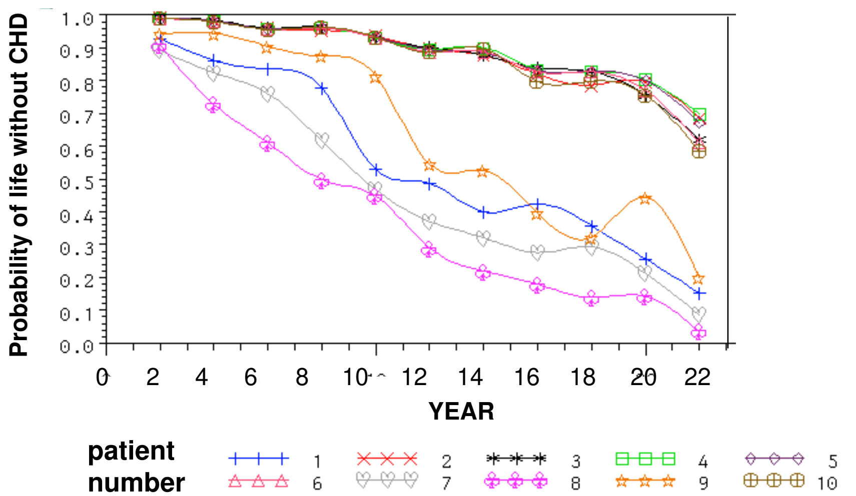
	Year									
	2		4		6		8		10	
Informative Year	*W*	*s.e.*	*W*	*s.e.*	*W*	*s.e.*	*W*	*s.e.*	*W*	*s.e.*
2			0.7311	0.0187	0.7244	0.0157	0.7275	0.0144	0.7331	0.0131
4	0.7243	0.0230			0.7169	0.0159	0.7282	0.0143	0.7311	0.0131
6	0.7314	0.0258	0.7312	0.0191			0.7318	0.0142	0.7376	0.0129
8	0.7317	0.0234	0.7401	0.0187	0.7302	0.0166			0.7339	0.0130
10	0.7390	0.0235	0.7230	0.0191	0.7354	0.0155	0.7325	0.0140		
12	0.7544	0.0231	0.7516	0.0181	0.7563	0.0153	0.7563	0.0135	0.7558	0.0128
14	0.7645	0.0232	0.7724	0.0179	0.7747	0.0154	0.7709	0.0137	0.7814	0.0126
16	0.7703	0.0233	0.7745	0.0179	0.7798	0.0157	0.7835	0.0138	0.7959	0.0129
18	0.7453	0.0247	0.7704	0.0177	0.7874	0.0154	0.7827	0.0142	0.7974	0.0133
20	0.7483	0.0236	0.7722	0.0179	0.7774	0.0159	0.7866	0.0143	0.8060	0.0132
22	0.7015	0.0256	0.7292	0.0200	0.7507	0.0175	0.7513	0.0166	0.7696	0.0159

	12		14		16		18		20	
Informative Year	*W*	*s.e.*	*W*	*s.e.*	*W*	*s.e.*	*W*	*s.e.*	*W*	*s.e.*
2	0.7448	0.0124	0.7724	0.0122	0.8065	0.0119	0.8322	0.0121	0.8363	0.0123
4	0.7455	0.0124	0.7721	0.0122	0.8045	0.0120	0.8301	0.0122	0.8369	0.0123
6	0.7430	0.0125	0.7756	0.0122	0.8073	0.0119	0.8304	0.0122	0.8347	0.0124
8	0.7414	0.0124	0.7733	0.0121	0.8033	0.0120	0.8309	0.0121	0.8364	0.0124
10	0.7444	0.0124	0.7689	0.0122	0.8070	0.0119	0.8336	0.0121	0.8358	0.0124
12			0.7742	0.0121	0.8046	0.0120	0.8293	0.0122	0.8359	0.0124
14	0.7730	0.0124			0.8060	0.0120	0.8324	0.0121	0.8366	0.0123
16	0.7954	0.0125	0.8058	0.0120			0.8265	0.0124	0.8332	0.0125
18	0.8052	0.0130	0.8248	0.0123	0.8337	0.0120			0.8337	0.0125
20	0.8127	0.0129	0.8301	0.0124	0.8420	0.0120	0.8404	0.0121		
22	0.7826	0.0159	0.8128	0.0153	0.8380	0.0144	0.8334	0.0152	0.8322	0.0158

	22	
Informative Year	*W*	*s.e.*
2	0.8413	0.0152
4	0.8437	0.0151
6	0.8415	0.0152
8	0.8448	0.0149
10	0.8432	0.0150
12	0.8343	0.0156
14	0.8315	0.0156
16	0.8367	0.0156
18	0.8411	0.0153
20	0.8444	0.0150

Figure 7.17 shows an example of ten survival curves produced by the sequential neural networks in which predictions in Year 20 were provided.

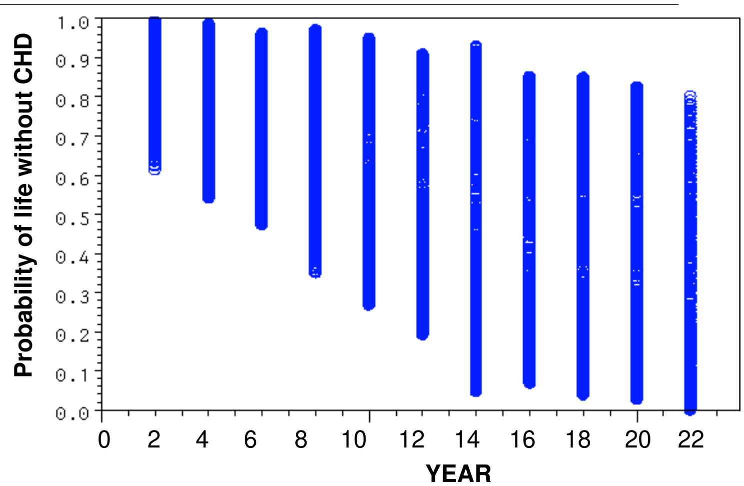
Figure 7.17. Survival curves using neural network and information from Year 20.



Nonmonotonic curves, such as the one corresponding to patient number 9, are still produced in sequential neural network models, but they are not as steep as in the standard neural network model.

Figure 7.18 shows the range of probabilities.

Figure 7.18. Range of probabilities for neural network model using information on Year 20.



The range of probabilities for sequential models is wide, including the ones produced for the first intervals.

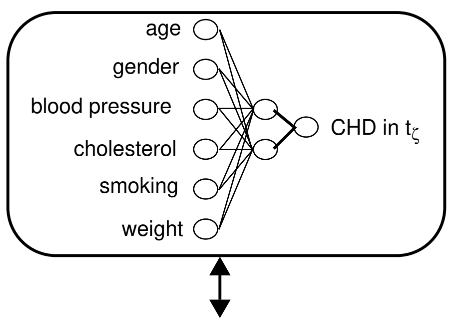
7.3 Model Comparison

In this section we compare neural networks and logistic regression models that use the same method (i.e., standard neural networks versus standard logistic regression models and sequential neural networks versus sequential logistic regression models). Comparison of resolution was simple, since there are published methods to compare pairs of areas under the ROC curves or the Wilcoxon statistic derived from the same cases [Hanley, 1983]. Overall resolution comparison across all models was done by a nonparametric sign test. Comparison of calibration is not as easily defined, and the results shown here report how many perfectly calibrated models were obtained in neural network and logistic regression models.

7.3.1 Standard neural networks versus standard logistic regression models

Equivalent standard logistic and standard neural network models, such the ones shown in Figure 7.19 were compared for calibration and resolution.

Figure 7.19. Standard neural networks and standard logistic regression models.



$$\text{Prob}(\text{CHD}_{t\zeta}) = \frac{e^{(a^*\text{age}_{t\zeta}+b^*\text{gender}+c^*\text{bp}_{t\zeta}+d^*\text{chol}_{t\zeta}+e^*\text{smoking}_{t\zeta}+f^*\text{weight}_{t\zeta})}}{1+e^{(a^*\text{age}_{t\zeta}+b^*\text{gende}_{t\zeta}\text{r}+c^*\text{bp}_{t\zeta}+d^*\text{chol}_{t\zeta}+e^*\text{smoking}_{t\zeta}+f^*\text{weight}_{t\zeta})}}$$

Standard neural networks perform the same task as that of standard logistic regression models. Results from both models are compared.

The results of applying the Hosmer-Lemeshow test to all standard models are shown in Figure 7.20. The perfect calibration curve was plotted for reference. Note that the scales change from graph to graph. Figure 7.21 shows the areas under the ROC curves.

Figure 7.20. Calibration plots and Hosmer-Lemeshow χ^2 (*p*) for all standard models.

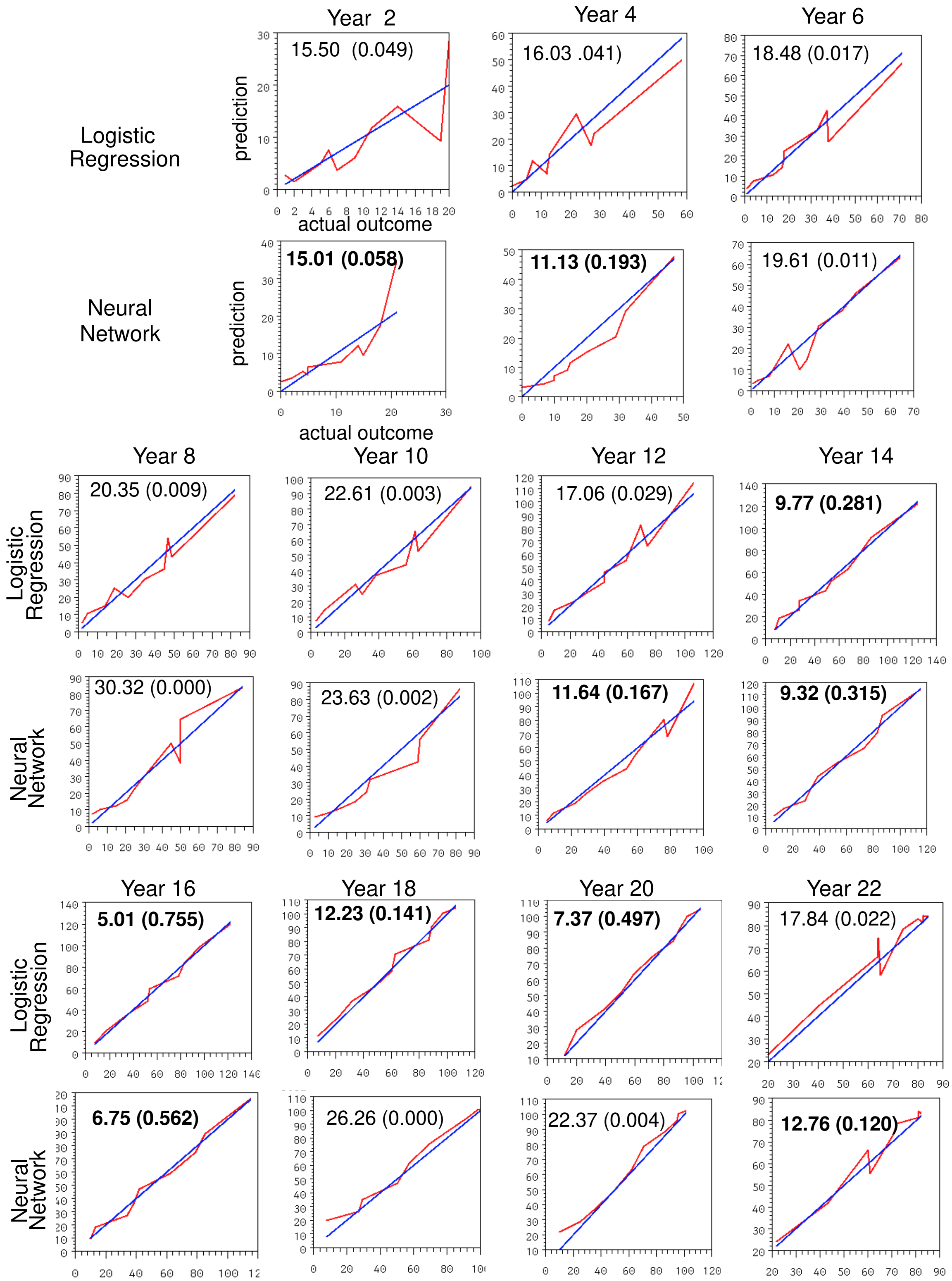


Figure 7.21. Areas under the ROC curves (standard errors) for all standard models

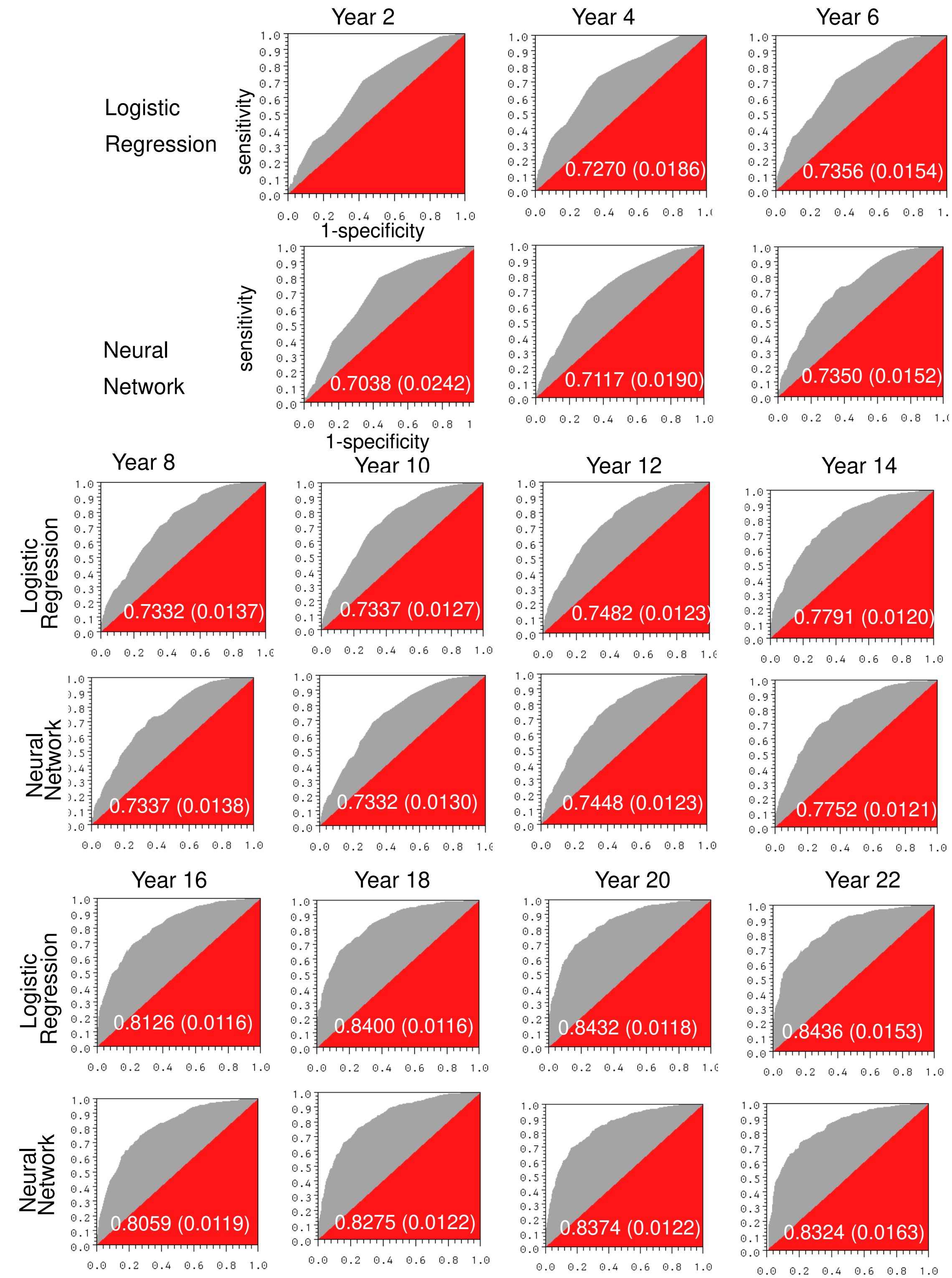


Table 7.11 shows the difference in resolution between the logistic regression and the neural network models and the p for testing the hypothesis that the differences are statistically significant for all intervals. There were no differences for $\alpha = 0.05$.

Table 7.11. **Differences (*d*) in resolution of standard models.**

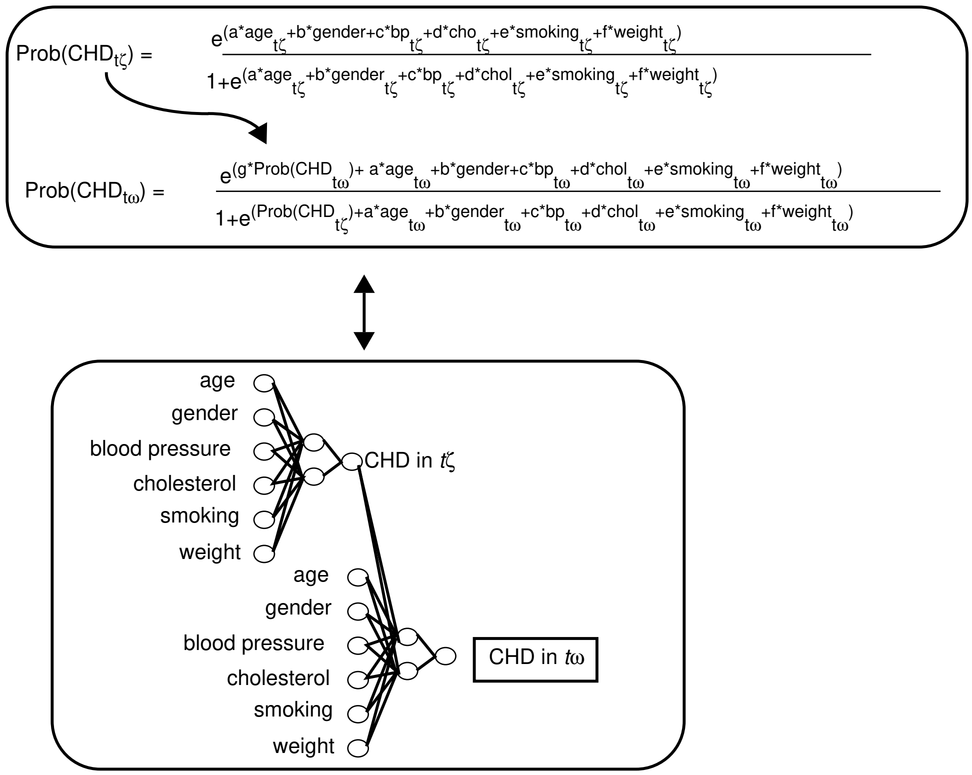
Year	*d*	*p*
2	-0.0320	0.94
4	0.0159	0.14
6	0.0004	0.48
8	-0.0004	0.51
10	0.0103	0.11
12	0.0034	0.34
14	0.0038	0.32
16	0.0067	0.20
18	0.0125	0.06
20	0.0057	0.22
22	0.0111	0.16

A rank sign test resulted in p=0.0674, suggesting the results were not statistically different.

7.3.2 Sequential neural networks versus sequential logistic regression models

Sequential logistic regression and sequential neural network models, similar to the ones shown in Figure 7.22, were compared for calibration and resolution.

Figure 7.22. Sequential logistic regression versus sequential neural network models.



Sequential logistic regression models perform the same task as that of sequential neural network models. Results are compared.

Table 7.12 shows the differences in resolution between the logistic regression and the neural network models for all intervals, as well as the p for testing the hypothesis that the differences were statistically significant. None of the differences were significant for α = 0.05.

Table 7.12. **Difference (d) in resolution of sequential models.**

	Year									
	2		4		6		8		10	
Informative Year	d	*p*	d	*p*	d	*p*	d	*p*	d	*p*
2			-0.0034	0.58	0.0088	0.23	0.0050	0.31	0.0104	0.13
4	-0.0112	0.70			0.0181	0.08	0.0053	0.30	0.0126	0.09
6	-0.0138	0.73	0.0011	0.47			0.0028	0.40	0.0066	0.23
8	-0.0007	0.51	0.0028	0.41	0.0170	0.08			0.0117	0.09
10	-0.0043	0.58	0.0242	0.07	0.0182	0.05	0.0099	0.15		
12	-0.0216	0.87	0.0018	0.44	0.0019	0.43	-0.0047	0.69	0.0010	0.45
14	-0.0138	0.77	-0.0032	0.60	0.0011	0.45	0.0010	0.46	0.0004	0.47
16	-0.0185	0.83	-0.0002	0.50	0.0077	0.24	0.0021	0.41	0.0041	0.31
18	0.0027	0.44	0.0048	0.36	0.0006	0.47	0.0068	0.24	0.0089	0.19
20	-0.0041	0.58	-0.0004	0.51	0.0109	0.19	0.0032	0.38	0.0027	0.38
22	0.0004	0.49	0.0093	0.29	0.0035	0.39	0.0005	0.48	0.0016	0.44

	12		14		16		18		20	
Informative Year	d	*p*	d	*p*	d	*p*	d	*p*	d	*p*
2	0.0037	0.31	0.0065	0.21	0.0064	0.22	0.0076	0.17	0.0065	0.22
4	0.0032	0.33	0.0071	0.19	0.0082	0.16	0.0098	0.13	0.0061	0.22
6	0.0054	0.27	0.0041	0.30	0.0057	0.24	0.0093	0.12	0.0084	0.17
8	0.007	0.18	0.0064	0.20	0.0099	0.11	0.0089	0.15	0.0066	0.20
10	0.0043	0.29	0.0108	0.10	0.0064	0.21	0.0058	0.23	0.0071	0.20
12			0.0053	0.24	0.0080	0.16	0.0098	0.14	0.0069	0.20
14	0.0073	0.19			0.0071	0.19	0.0068	0.21	0.0065	0.22
16	0.0105	0.10	0.0039	0.32			0.0135	0.07	0.0103	0.12
18	0.0143	0.06	0.0058	0.22	0.0072	0.17			0.0095	0.13
20	0.0100	0.12	0.0059	0.23	0.0057	0.25	0.0078	0.18		
22	0.0075	0.26	0.0001	0.49	-0.0023	0.59	0.0107	0.15	0.0050	0.32

	22	
Informative Year	d	*p*
2	0.0020	0.42
4	-0.0003	0.51
6	0.0024	0.41
8	-0.0014	0.55
10	-0.0009	0.53
12	0.0079	0.26
14	0.0110	0.19
16	0.0057	0.32
18	0.0025	0.41
20	-0.0012	0.54

A rank sign test to test the hypothesis that the differences were not significant resulted in p=0.0001, favoring the logistic regression model.

7.4 Method Comparison

7.4.1 Standard versus sequential logistic regression

Standard and sequential logistic regression models, similar to the ones shown in Figure 7.23, were compared for calibration and resolution.

Figure 7.23. Standard versus sequential logistic regression: Framingham data.

$$\mathrm{Prob}(\mathrm{death}_{t\zeta}) = \frac{e^{(a*age_{t\zeta}+b*gender_{t\zeta}+c*bp_{t\zeta}+d*chol_{t\zeta}+e*smoking_{t\zeta}+f*weight_{t\zeta})}}{1+e^{(a*age_{t\zeta}+b*gender_{t\zeta}+c*bp_{t\zeta}+d*chol_{t\zeta}+e*smoking_{t\zeta}+f*weight_{t\zeta})}}$$

$$\updownarrow$$

$$\mathrm{Prob}(\mathrm{CHD}_{t\zeta}) = \frac{e^{(a*age_{t\zeta}+b*gender_{t\zeta}+c*bp_{t\zeta}+d*chol_{t\zeta}+e*smoking_{t\zeta}+f*weight_{t\zeta})}}{1+e^{(a*age_{t\zeta}+b*gender_{t\zeta}+c*bp_{t\zeta}+d*chol_{t\zeta}+e*smoking_{t\zeta}+f*weight_{t\zeta})}}$$

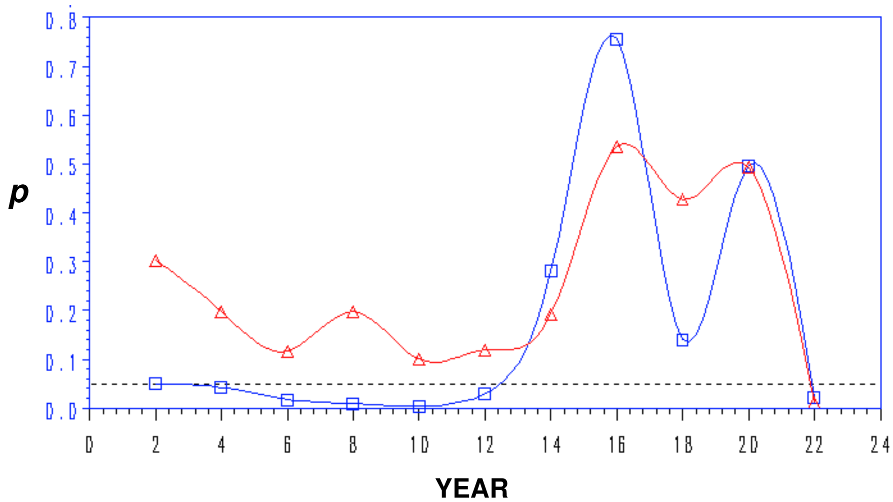
$$\mathrm{Prob}(\mathrm{CHD}_{t\omega}) = \frac{e^{(g*Prob(CHD_{t\zeta})+\ a*age_{t\omega}+b*gender_{t\omega}+c*bp_{t\omega}+d*chol_{t\omega}+e*smoking_{t\omega}+f*weight_{t\omega})}}{1+e^{(Prob(CHD_{t\zeta})+a*age_{t\omega}+b*gender_{t\omega}+c*bp_{t\omega}+d*chol_{t\omega}+e*smoking_{t\omega}+f*weight_{t\omega})}}$$

Standard and sequential models were compared for calibration and resolution.

The comparison of the Akaike information criterion showed that the sequential logistic regression models had a better fit to the data than the standard regression models.

Figure 7.24 shows the p obtained by the Hosmer-Lemeshow test on the standard models and the average p obtained by the same test on the sequential models. Overall, there was small change in calibration favoring the sequential model, especially for the first intervals.

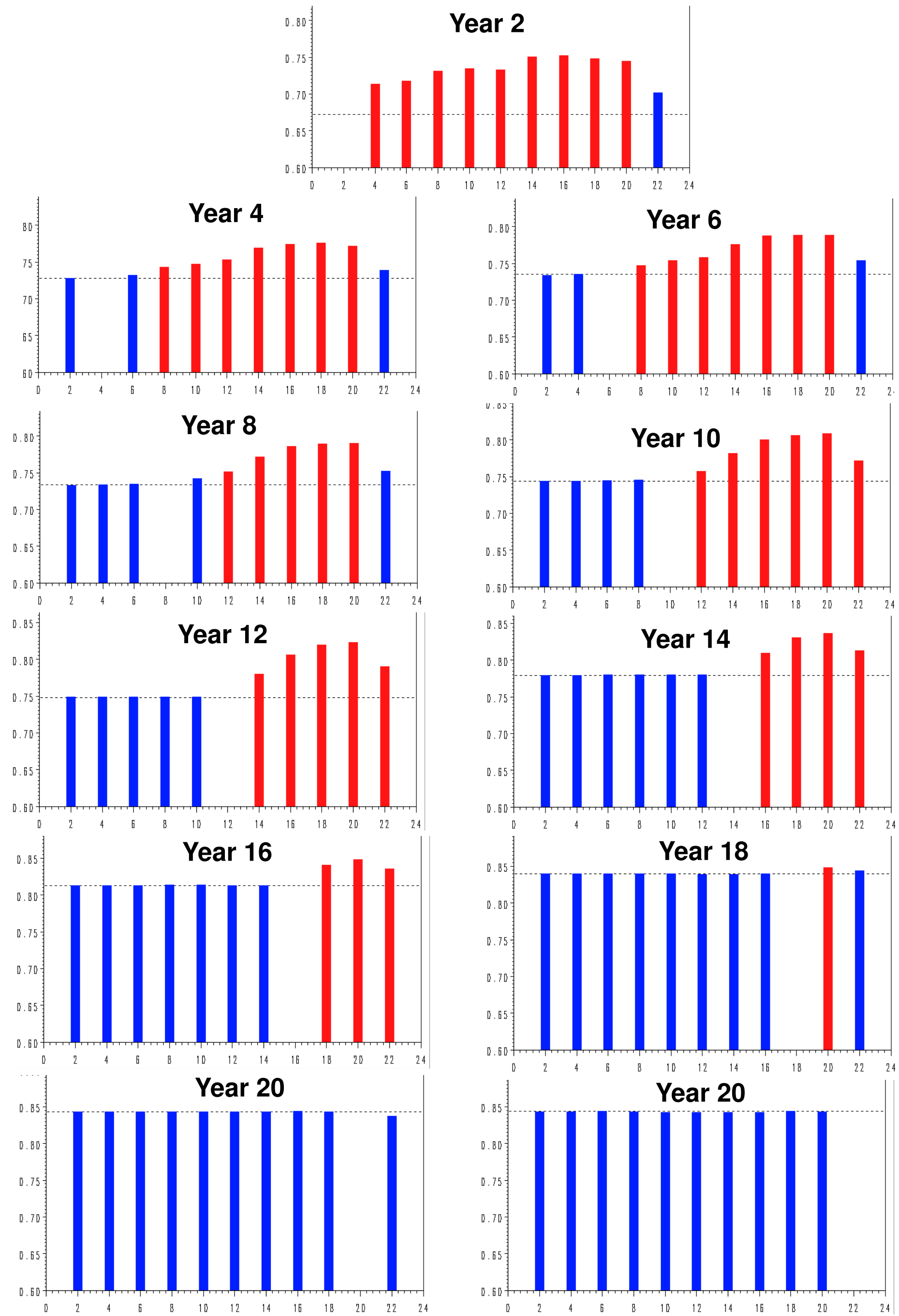
Figure 7.24. Calibration of standard and sequential logistic regression models.



The *p* (triangles) and average *p* (squares) obtained by the Hosmer-Lemeshow test are plotted for the standard and sequential models, respectively. There was not a small change in calibration favoring the sequential method.

Of the 110 sequential logistic regression equations, 47 resulted in significantly larger areas under the ROC curve (p<0.05) than their equivalent standard equations. None of the sequential equations produced areas that were statistically smaller than their standard counterparts, as shown in Figure 7.25. The ROC curves produced by the sequential logistic regression models had higher standard errors than those of the standard models because the number of cases was either the same or smaller.

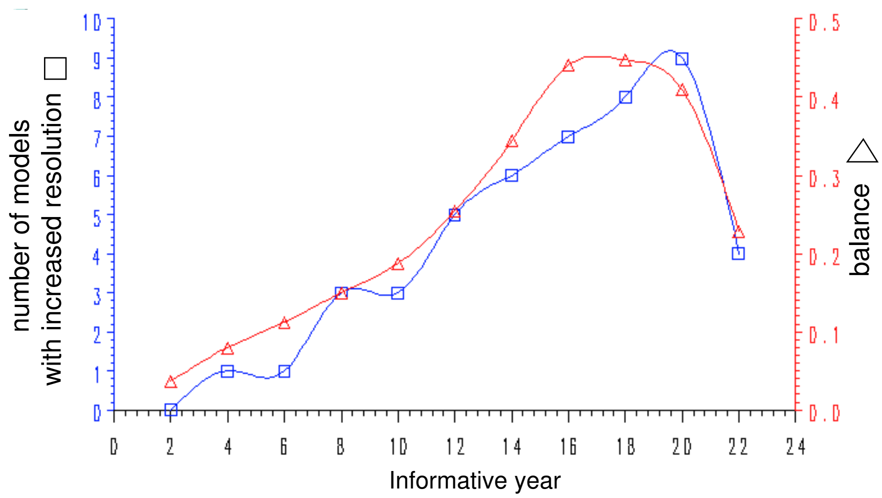
Figure 7.25. Resolution of standard and sequential logistic regression models.



Bars represent the area under the ROC curve for sequential models when different informative years were entered in the model. The dotted line represents the resolution of the standard model for reference.

Consider each pair of *(A,B)*, where *A* is the year for which predictions are produced in the sequential logistic regression, and *B* is the year whose predictions from the standard model were provided to build the sequential model. The “most informative year” is defined as the most frequent value of *B* in the 47 *(A,B)* pairs. This value was 20. The “most informed year,” that is, the year that most benefitted from information on another year is defined as the most frequent *A* in the same 47 pairs. This value was 2. Conversely, the “least informative year” is defined as the most infrequent value of *B* in the 47 *(A,B)* pairs, and was 2. The “least informed year” is defined as the most infrequent value of *A* in the 47 pairs and was 20. Note, in Figure 7.26, that the degree to which each year was informative is correlated with the balance of the data: The more balanced the data were, the more information the predictions for that year could enhance resolution.

Figure 7.26. Informative years and balance in sequential logistic regression model.

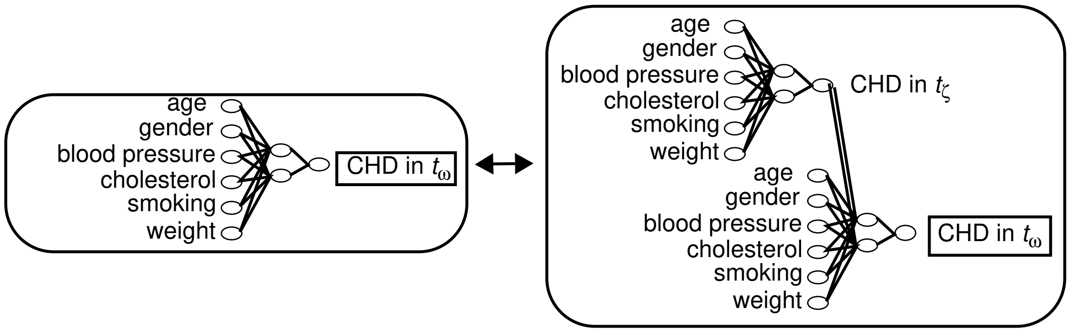


Squares represent the number of times an informative year corresponded to a significant improvement in the area under the ROC curve and are scaled in the left axis. Triangles represent the data balance and are scaled in the right axis. The “most informative years” were the ones in which data were most balanced.

7.4.2 Standard versus sequential neural networks

Standard and sequential neural network models, similar to the ones shown in Figure 7.27, were compared for calibration and resolution.

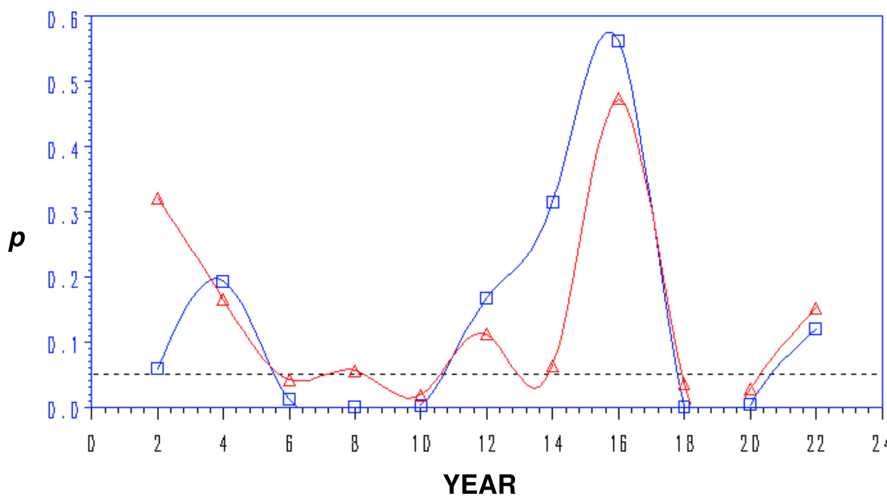
Figure 7.27. Standard versus sequential neural network: Framingham data.



Standard and sequential models were compared for calibration and resolution.

Figure 7.28 shows the *p* obtained by the Hosmer-Lemeshow test on the standard models and the average *p* obtained by the same test on the sequential models. Overall, there was not a major change in calibration: it stayed low in intervals where it was low, and did not have major increases in intervals where it was already high.

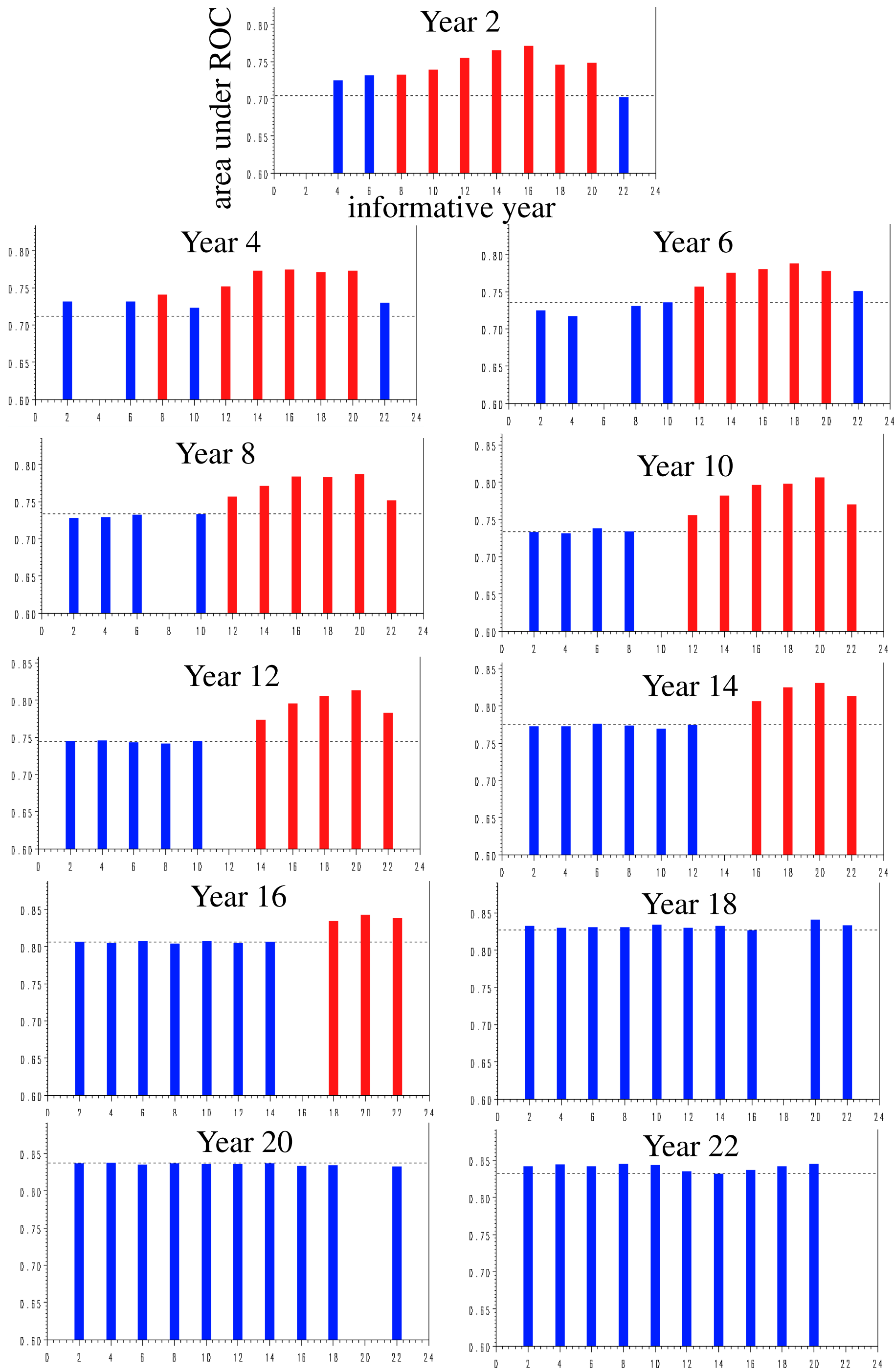
Figure 7.28. Calibration of standard and sequential neural network models.



The *p* (triangles) and average *p* (squares) obtained by the Hosmer-Lemeshow test are plotted for the standard and sequential models, respectively. There was not a major change in calibration when the sequential method was used.

Of the 110 sequential neural networks, 42 resulted in significantly larger areas under the ROC curve ($p<0.05$) than their equivalent standard models, as shown in Figure . None of the sequential networks produced areas that were statistically smaller than their standard counterparts.

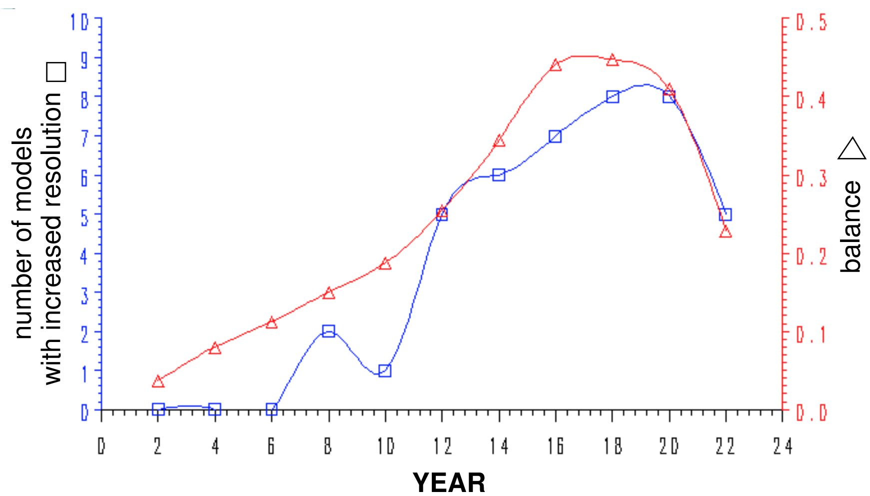
Figure 7.29. Resolution of sequential neural networks.



Bars represent the area under the ROC curve for sequential models when different informative years were entered in the model. The dotted line represents the resolution of the standard model for reference.

The “most informative year” was 20. The “most informed year” was 2. The “least informative year” was 2. The “least informed year” was 20. Note, in Figure 8.31, that here the degree to which each year was informative is also positively correlated with the balance of the data.

Figure 7.30. Informative years and balance in sequential neural network models.



Squares represent the number of times an informative year corresponded to a significant improvement in the area under the ROC curve and are scaled in the left axis. Triangles represent the data balance and are scaled in the right axis. The “most informative years” were the ones in which data were most balanced.

7.5 Summary of Results

No difference in resolution could be detected for standard logistic regression and sequential neural network models. Standard logistic regression and standard neural network models were well calibrated in different intervals (six out of eleven times calibration of standard logistic regression models was superior).

No direct pairwise difference in performance in terms of resolution could be detected for sequential logistic regression and sequential neural network models, although an overall sign test indicates statistical difference favoring the sequential logistic regression models.

The sequential method performed better than the standard method, in terms of calibration and resolution, for both logistic regression and neural network models in this data set.

7.6 Discussion

It is surprising that the neural network models did not outperform their logistic regression counterparts. As discussed previously in Chapter 2, neural networks with hidden layers can approximate more functions than can logistic regression models. There may be several reasons why, in this particular problem, the neural networks models did not provide any advantage over logistic regression models:

(1) The number of hidden nodes may have been insufficient, constraining the number of functions that could be approximated by the neural networks. Although I tried neural networks with more hidden nodes (e.g., 30 hidden nodes), and did not get results that were significantly different from the ones shown here, I might have tried an even higher number. However, the error in the holdout set started to increase after a certain number of cycles, indicating that the number of hidden nodes probably did not play a pivotal role in limiting the performance.

(2) The learning rate was fixed and, perhaps, inadequate. It is possible that the choice of too large a learning rate made it impossible for the network to reach a state of minimal error. There are currently no guidelines on how to choose the ideal learning rate for a specific problem, other than the general knowledge that the learning rate should be diminished when the network is incapable of learning, which was definitely not the case in this experiment.

(3) The logistic regression models had an excellent fit. Indeed, the Akaike information criterion for every logistic regression model tested in this experiment indicated that there was a good fit to the data. Neural networks should be able to easily approximate the

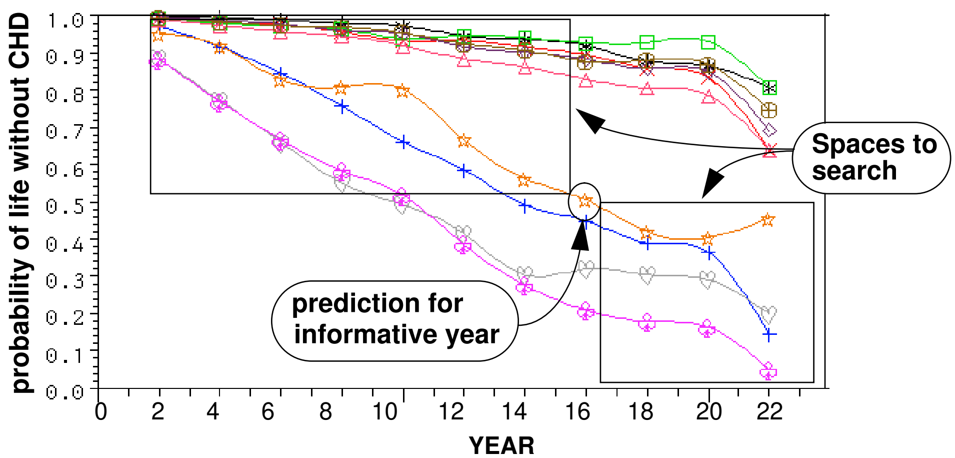
logistic function, but it is hard to detect overfitting when the backpropagation algorithm is used in a multilayered neural network. The criterion used to stop training (stop when the error in the holdout set was twice that of the error in the best cycle so far) may have been inapropriate. Furthermore, the need for a holdout set restricted the effective size of the training set.

It is not surprising to see that sequential methods outperform standard methods. In this particular application of logistic regression and neural networks for survival analysis, an important piece of information is missing when prognostic indices are created for each interval: the model does not have the commonsense knowledge to know that the results of one interval are dependent on the results of other intervals. This is why nonmonotonic curves are created. Although the sequential method still allows nonmonotonicity, there are fewer instances when it in fact occurs.

The fact that predictions coming from intervals where data was more balanced (year 20) were more informative was expected. The fact that the highest increases in the area under the ROC curve occurred for intervals where the data was less balanced (year 2) was also not a surprise. It was unclear, however, whether the improvement in resolution would imply a decrease in calibration. As discussed previously in Chapter 6, it is hard to directly compare calibrations for these two methods, but there was apparently no decrease in this case. The range of probabilities produced by sequential models was larger than that of standard models, especially for the less balanced intervals, suggesting even an improvement in calibration. A larger range of probabilities for the interval corresponding to Year 2, for example, means that the models were able to produce overall smaller values for their predictions or CHD development in that year, being able to (a) better approximate the probability of developing CHD (better calibration), and (b) better differentiate between patients who developed and who did not develop CHD in that interval (better resolution). In all cases in which there was a significant difference in resolution between the standard and the sequential methods, this difference favored the latter. Even among the nonsignificant differences, the majority favored the sequential method, suggesting that, everything else being equal, it should be the first choice for researchers dealing with prognostic indices for time-oriented dependent data.

Sequential methods are probably facilitating the assignment of probabilities for each interval by constraining the universe of possibilities for a given probability value in intervals surrounding the informative interval, as illustrated in Figure 7.31.

Figure 7.31. Sequential models and limitation of search space.



Sequential models seem to be facilitating the task of assigning prognoses by limiting the search space of possible probability values at certain intervals.

In this study, there was never a statistically significant difference for models in which the informative year preceded the year for which the prognostic index was produced. For example, it was never the case that predictions for year 2 accounted for a significant difference in performance for any other interval, predictions for year 6 were only informative to models predicting CHD in 2 or 4 years, etc. In other data sets, this should not always be the case. For example, in domains where all patients happen to have the same final outcome after a certain time (e.g., death), the last intervals may be as unbalanced as the first ones, since the event is nonreversible and it will have happened to most patients in those intervals. For predictions in these last intervals, it is probably the case that the sequential method will also be useful, and predictions for an interval that preceded the ones in question may be helpful. In this data set, it was not possible to verify this intuition.

This study also suggests that no improvement in resolution can be obtained if the resolution of the informative interval is lower than that of the interval in question, regardless of data balance (e.g., data was unbalanced for year 22, and yet no improvement was achieved with any sequential models, including the ones in which predictions for the highly balanced years 16 or 18 were provided). Evidently, there is an upper bound on the resolution that can be obtained with any interval in a data set. Once this level is achieved, no more information can be extracted from the data.

Although resolution was almost identical for logistic regression and neural network models, calibration was not. For some intervals, calibration of logistic regression models was better than that of neural networks; for others, the reverse was true. This result indicates that both models may be determining the probability of CHD development by different means, suggesting that a combination of models (e.g., a mixed model in which results from one model are entered into another) may be useful. Since neural networks allow a larger number of functions to be modeled than logistic regression models, predictions obtained from the latter should constitute additional imputs to the former.

CHAPTER 8

Models of Survival in HIV Infection

This chapter describes the experiments that were conducted, using the ATHOS data set, to compare the performance of Cox proportional hazards, standard neural networks, and sequential neural networks in the prediction of death related to AIDS. These models were used to test the hypotheses that (a) sequential systems of neural networks perform better than standard neural networks, and (b) neural networks produce better estimates of survival time than Cox proportional hazards models.

Section 8.1 gives an overview of the problem and significance, emphasizing the need for predictive models for prognosis of AIDS and commenting on the deficiencies of current models. Section 8.2 describes the Cox proportional hazards model and the neural network model used in this experiment. Section 8.3 compares the performances of the Cox model and the standard neural network, showing that standard neural networks produce more accurate estimates of survival than those of the Cox model. Section 8.4 compares the performances of standard versus sequential neural networks, showing that sequential neural networks enhance the resolution of standard neural network models. Section 8.5 summarizes the results, and Section 8.6 discusses their implications to AIDS modeling.

8.1 Prognosis of Death Due to AIDS: Existing Models

HIV infection is the most challenging public health problem that has arisen in the second half of the twentieth century. Worldwide, AIDS has killed more than 300,000 people in the last decade, and HIV has infected 5 to 10 million people [Quinn, 1990]. It is, therefore, vitally important to understand the natural history of the disease, to be able to predict the development of the disease in HIV+ individuals, to formulate national policies to slow disease progression, and to establish guidelines for adequate medical intervention. Modeling state transitions from HIV+ status to AIDS and from AIDS to death provides tools for the health care provider to use in predicting the prognoses of HIV+ patients and in formulating adequate health care policies.

Prognostic evaluations of patients who are HIV+ can help both patients and physicians to allocate resources. The classification of patients according to different disease-progression profiles also helps policymakers to determine the nation's health care needs. Escalating health care costs related to the prevention and treatment of HIV infection call for policies that are derived from existing data using reliable quantitative tools [Seage, 1990]. In this project, I built and evaluated a Cox proportional hazards model and a neural network model to predict the survival of patients who had AIDS according to the 1993 CDC definition of the disease [Centers for Disease Control and Prevention, 1993] in a specific data set.

Several authors have shown the relevance of demographic, physical, biochemical, and therapeutic factors in the development of AIDS. The demographic factors include age [Easterbrook, 1993; Lemp, 1990; Rothemberg, 1987], gender [Rothemberg, 1987], ethnicity [Easterbrook, 1993 and 1991; Moore, 1991; Rothemberg, 1987], and risk group [Rothemberg, 1987]. Initial presentation of the immunodeficiency is also a marker related to prognosis, especially if oral thrush [Lin, 1993; Rabeneck, 1993; Selwyin 1992; Saah, 1992], esophageal candidiasis [Hanson, 1993], *pneumocystis carinii* pneumonia [Chang, 1993; Harris, 1990], Kaposi's sarcoma [Seage, 1993; Levine, 1991], pyogenic bacterial infections [Alcabes, 1993], or atypical mycobacterial infections [Wenger, 1988] are present. Physical factors affecting state transitions include constitutional symptoms and physical signs, such as fatigue [Saah, 1992], clinical anemia [Gardner, 1992], weight loss [Hanson, 1993; Chlebowski, 1989], and diarrhea. Laboratory markers of disease progression include CD4 [Rubsamen-Waigmannm 1991; Reibnegger, 1991; Fahey, 1990; Schechter, 1990; Pedersen, 1989], white blood cell and platelet counts [Justice, 1989], CD4/CD8 ratio, serum p24 antigen [Fahey, 1990; Pedersen, 1989], HIV viremia, hemoglobin [Justice, 1989; Fuchs, 1989], HDL [Rubsamen-Waigmann, 1991], albumin [Justice, 1989; Chlebowski, 1989], serum IgA [Schechter, 1990; Pedersen, 1989], serum β-2 microglobulin levels [Whittle, 1992], erythrocyte sedimentation rate [Hanson, 1993], and skin tests. Therapeutic factors influencing the prognosis of the HIV infection include prophylaxis for specific opportunistic infections, such as *pneumocystis carinii* pneumonia and infection by atypical mycobacteria, and antiretroviral therapy [Graham, 1991]. The least controversial marker is the CD4 lymphocyte count. A review of existing markers for disease progression in HIV+ people was done by Libman [1992]. Curtis et al. [1993] reviewed the studies relating ethnic factors and survival time with AIDS.

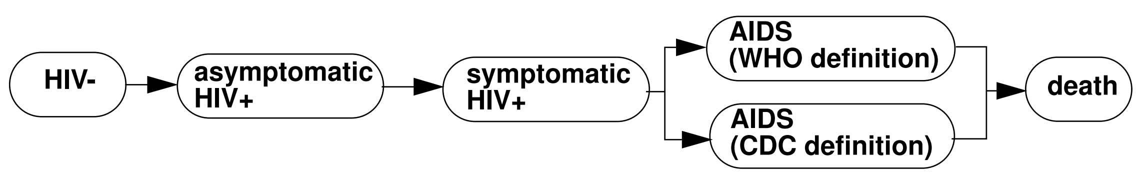
I used common markers of disease progression available in the ATHOS data set to make predictions of death for patients who had AIDS. This study was intended to assess predictive performance of different models that delineate patterns of survival for different patients in the ATHOS data set, to show how neural networks can be useful for modeling survival in this population, and to compare predictive performance in test cases deriving from the same population. Results in other populations may be very different. As we will see in Section 8.2, variables indicating interventions, such as antiretroviral therapy, were entered as independent variables. If these variables reflect the influences of unknown variables (not included in the study), and in a different population the reasons for determining an intervention are not exactly the same, or the correlations between the unmeasured variables and the treatment variables follow a different pattern, then the results obtained in this sutdy will not be reproduceable. On the other hand, if the interventions are based solely on the variables already entered in the sudy, collinearity may be a problem. This study was not intended to establish causal relations, but rather predict outcomes based on the observed variables.

The ATHOS data set is described in Section 8.2.1. A full account of the markers utilized in the experiments is given later in this chapter, in Section 8.2.2. Before describing the experiments in detail and justifying the choice of neural networks as a modeling tool, however, I will comment on existing models of disease progression in HIV infection and their advantages and disadvantages.

8.1.1 Nonparametric models for disease progression

Several authors have modeled transitions from seronegative to seropositive HIV, from asymptomatic HIV+ status to symptomatic HIV+ status, from HIV+ status to AIDS, and from AIDS to death [Mariotto, 1992; Longini, 1989], as shown in Figure 8.1.

Figure 8.1. Transitions from HIV- to death.



It is possible to model each transition from HIV- state until death. Each transition requires different assumptions and has a corresponding error associated with it. Earlier transitions are more difficult to model because the precise date of infections is usually unknown, and the occurrence of the first symptoms cannot be easily determined.

The simplest nonparametric models for disease progression use actuarial life tables or Kaplan–Meier product-limit estimators (see Section 5.2). Applications of both models in the domain of HIV infections have been published [Aragon, 1993; Moore, 1992; Dannemann 1992; Fischl, 1987; Vadhan-Raj, 1986]. In the domain of HIV infection, nonparametric models such as classification trees have been used to model survival [Segal, 1989; Piette, 1992]. Parametric models in the domain of AIDS have also been used [Mariotto, 1992; Longini, 1991 and 1989].

The Cox proportional hazards model is classified as a semiparametric model, and involves the assumptions that there is a simplifying transformation of the initial data and that the hazards for the different groups are proportional, as discussed in Chapter 5 [Selvin, 1991]. Hanson et al. [1993] found that the proportional assumption required by the Cox proportional hazards model did not hold for their cohort of HIV+ patients. Under these circumstances, a nonparametric predictive model, such as a neural network, may be more appropriate. Few studies have addressed the predictive power of neural networks in survival analysis, as discussed in Chapter 5. Neural networks have rarely been used to model survival in the domain of HIV infection, and their comparison to other prognostic models has been limited. A particular implementation of Cox proportional hazards to model death for AIDS patients is shown in Section 8.2.3.

8.2 Experimental Design and Results

Cox proportional hazards and neural network models were built to make predictions of death for patients who had AIDS, and to produce individualized survival curves for these patients. Standard and sequential systems of neural networks were used. All models were built using the ATHOS data set, and the comparisons were made on the same subset of cases.

8.2.1 The ATHOS data set

The ATHOS database is a longitudinal, primary data set of HIV+ and at-risk subjects, collected from 10 clinics in California (3 private practices in the San Francisco Bay Area, 2 private practices in Los Angeles, and 5 community clinics associated with the Owen Clinic at University of California–San Diego), under the direction of Dr. James Fries [1992]. The ATHOS database was built to provide a national HIV data resource that permits the systematic study of (1) disease costs and financing, (2) drug effectiveness, toxicity, and cost, (3) delivery systems and practice variations, (4) health status and quality of life, and (5) disease transitions and modeling. The data collection began in 1989 at

Stanford University and involved abstraction of existing charts, as well as prospective collection of clinical and laboratory information. Some authors have used the ATHOS database to assess the socioeconomic impact of the AIDS epidemic, as well as to investigate medical issues related to HIV infection [McShane, 1993; Lubeck 1993 and 1992].

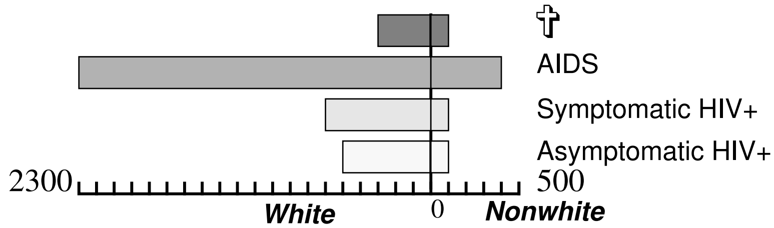
Data from 5471 patients were available for mortality studies and over 700 variables were represented. The variables included diagnoses, signs and symptoms, results of laboratory tests, and medications. Additional detailed data, collected from questionnaires, were available for 1335 of the patients. Variables included in the questionnaire assess functional status, quality of life, insurance coverage, medical resources utilization, side effects, and multiple health outcomes. Data were collected in three-month intervals. The distribution of cases in the various ethnic groups is shown in Table 8.1.

Table 8.1. Distribution of cases according to ethnicity

Ethnic group	Percentage of cases
White	85%
Hispanic	8%
African-American	4%
Asian or Pacific Islander	1%
Native American	1%
Other origins	1%

Researchers of the ATHOS project have developed quality-control protocols that assure the reliability of ATHOS data. Approximately 50 percent of the patients have AIDS (1993 CDC definition), 25 percent are HIV+ but do not have AIDS, and 25 percent are HIV-, but at risk for HIV infection. There were 290 deaths and 572 diagnoses of AIDS through mid-1993. Figure 8.2 shows the distribution of patient cases according to clinical stage and ethnicity.

Figure 8.2. Distribution of ATHOS patients according to clinical stage and ethnicity.



Some patients enrolled in the ATHOS study do not have AIDS.

A subset of the ATHOS data set was used for the experiments described in this chapter. Not all AIDS patients from the ATHOS data set were used because some lacked the date of AIDS diagnosis. Table 8.2 shows the distribution of cases according to the year of follow-up. Censored cases were not used in the models.

Table 8.2. **Distribution of cases according to year of follow-up.**

Year of follow-up	Dead	Alive	Total	Balance
1	64	850	914	0.0752
2	150	606	756	0.1984
3	229	358	587	0.3901
4	257	199	456	0.4364
5	274	86	360	0.2388
6	277	28	305	0.0918

8.2.2 Specification of covariates and outcomes

The major endpoint in this analysis was prediction of mortality due to AIDS-related conditions. Survival was measured from the date of AIDS diagnosis using the 1993 CDC definition. Variables were included in the model only when the literature showed that they have been proven to be informative. Not all published markers for disease progression in HIV infection were available in the ATHOS data set. Only baseline values, at the time of AIDS diagnosis, were used. No time-dependent variables were used.

Demographic and socioeconomic explanatory variables included age, gender, race, risk group, AIDS-defining diagnoses, insurance coverage, hospitalizations, and time elapsed from the estimated HIV seroconversion. Clinical findings included fatigue, weight loss, diarrhea, mental status, and Karnofsky scores. Laboratory test results included CD4 counts, CD4/CD8 ratio, hemoglobin, erythrocyte sedimentation rate, erythrocyte and platelet counts, white blood cell counts, serum p24 antigen, serum β-2 microglobulin, total cholesterol, HDL, and albumin levels. Variables indicating antiretroviral and prophylactic medications for opportunistic infections were also entered, as well as AIDS-related conditions reported after the patient entered the study. Continuous variables were represented as such, but they were normalized before entry. Dummy coding was used for categorical variables. Table 8.3 displays all independent variables. The dependent variable was death due to AIDS.

Table 8.3. **Independent variables.**

Demographic	**Clinical**	**Laboratory**	**Interventions**
Age	Fatigue	CD4 count	Antiretroviral therapy*
Gender	Weight loss	CD4/CD8	Prophylactic therapy for opportunistic infections
Risk group*	Diarrhea	hemoglobin	Therapeutic medications
AIDS-defining diagnosis*	Mental status	ESR	
Time elapsed from HIV sero-conversion	Karnofsky score	Erythrocyte count	
Length of stay in hospital	AIDS-related dis-orders*	Platelet count	
		WBC count	
		p24 antigen	
		Total cholesterol	
		HDL cholesterol	
		Albumin	
		β2-microglobulin	

* Dummy-coded variables

8.2.3 Cox proportional hazards model

A Cox proportional hazards model, simplified in Figure 8.17, was built. The SAS/STAT procedure PHREG with its default parameters was used to build the model. The BASELINE statement was used to produce a survival curve for each patient.

Figure 8.3. Cox proportional hazards: ATHOS data.

$$\frac{h(t)}{h_o(t)} = e^{(a*age+b*gender+c*Karnofsky+d*CD4+e*p24+f*antiretroviral\_therapy+...)}$$

In a Cox proportional hazards model, the hazard at time *t, h(t)*, is related to a baseline hazard $h_0(t)$ by an exponential function. The regression model is given the survival time for a given patient and calculates the proportional hazards for all time intervals. If a baseline hazard is provided, the hazard (and the survival) can be easily calculated.

Table 8.4 shows the calibration of the Cox proportional hazards model for predictions in one to six years. Note that, except for year 6, predictions were poorly calibrated.

Table 8.4. **Calibration of Cox proportional hazards model.**

Year of follow-up	χ^2	*p*
1	24.0970	0.0022
2	22.3187	0.0043
3	25.7879	0.0011
4	26.1228	0.0010
5	24.4776	0.0019
6	2.61953	0.9559

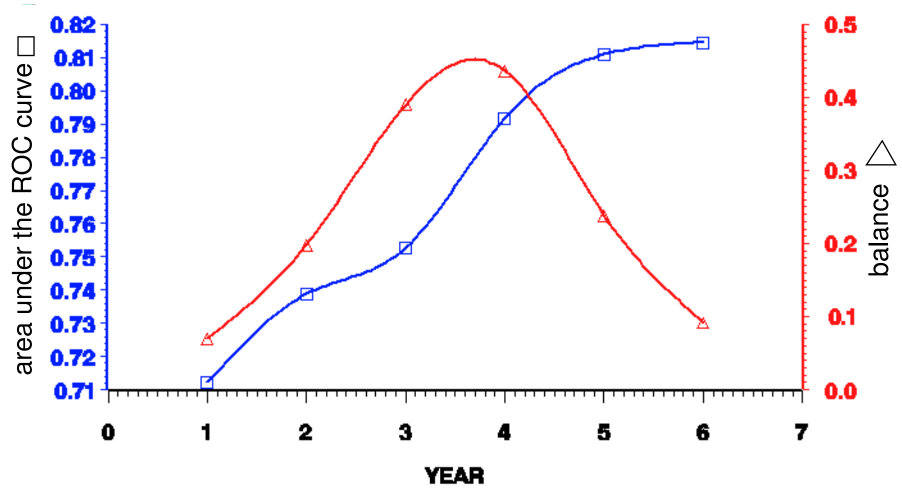
Table 8.2 shows the resolution of the model.

Table 8.5. **Resolution of Cox proportional hazards model.**

Year of follow-up	Area under the ROC curve	Standard error
1	0.7122	0.0342
2	0.7388	0.0239
3	0.7527	0.0213
4	0.7917	0.0207
5	0.8109	0.0250
6	0.8145	0.0323

The best discrimination was achieved for a follow-up of six years. Figure 8.4 shows how resolution is related to data balance.

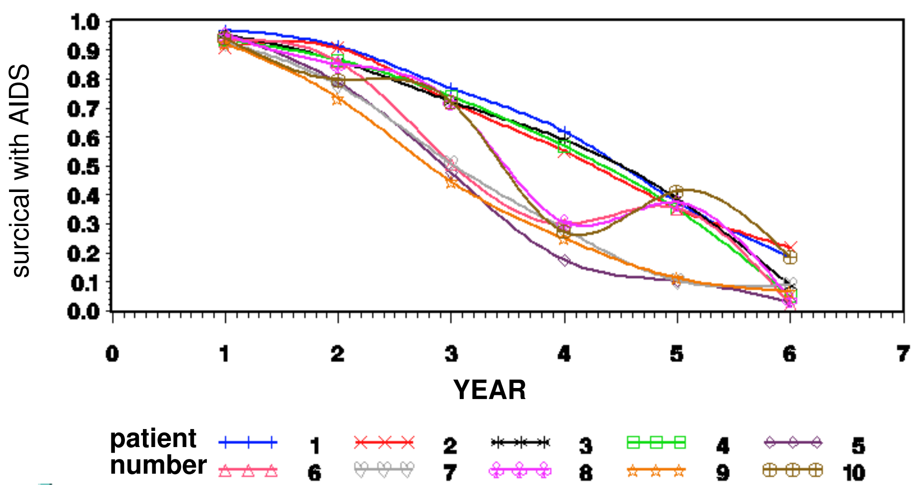
Figure 8.4. Resolution and data balance in the Cox proportional hazards model.



Resolution is represented by squares and scaled in the left axis. Balance is represented by triangles and scaled in the right axis.

Figure 8.5 shows an example of 10 survival curves produced by the Cox proportional hazards model.

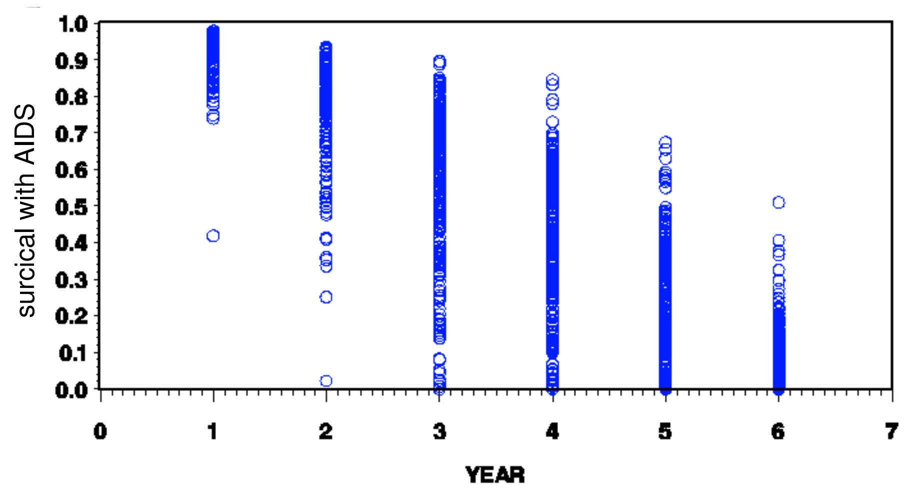
Figure 8.5. Survival curves for 10 patients using the Cox proportional hazards model.



Since the models have no relation to each other, survival curves that are not monotonically decreasing, although impossible in theory, can be produced, such as for patient number 10.

The range of probabilities produced by the model in the first and last intervals was very limited (most predictions would correspond to values over 0.75), as shown in Figure 8.6.

Figure 8.6. Range of probabilities for the Cox proportional hazards model.



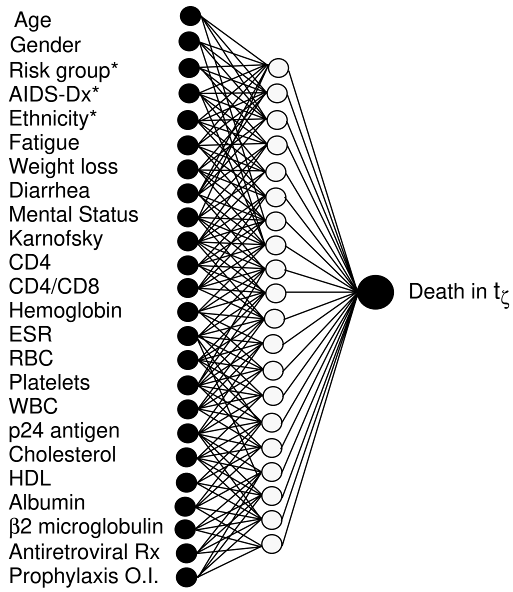
Predictions for the first and last intervals tended to be conservative, avoiding extreme values. Predictions for the intermediate intervals were better distributed, indicating a better data balance.

The range of probabilities in the intermediate intervals (e.g., three and four years) was wider, indicating a higher balance of people in those intervals. Evidently, the balance for the first and last intervals is lower, since in the first intervals almost all patients are alive, and in the last intervals almost all patients are dead. It is not surprising that the predictions are more spread in intervals where the balance is high (years 3 and 4) than in intervals where the balance is low (years 1 and 6).

8.2.4 Standard neural network model

A standard neural network model, similar to the one shown in Figure 8.7, was built.

Figure 8.7. Standard neural network: ATHOS data.



Variables marked by * are composed of several binary variables. For example, *Risk group* is composed of *Gay/Bisexual*, *IVDU*, *Heterosexual,* and *Transfusion recipient*. *AIDS-Dx* is composed of *PCP pneumonia*, *Kaposi sarcoma*, and so on.

The neural network had 38 inputs and 20 hidden nodes. It was trained by backpropagation with adaptive learning rate. Overfitting was monitored in a holdout set of 40 percent of the cases. The software package NevProp2 [Goodman, 1994] was used.

Table 8.2 shows the calibration of the standard neural network model for each year. Except for year 3, calibration was good.

Table 8.6. **Calibration of standard neural network models.**

Year of follow-up	χ^2	***p***
1	7.78301	0.45495
2	8.62868	0.37458
3	25.7896	0.00114
4	9.28662	0.31870
5	10.7534	0.21607
6	12.3251	0.13728

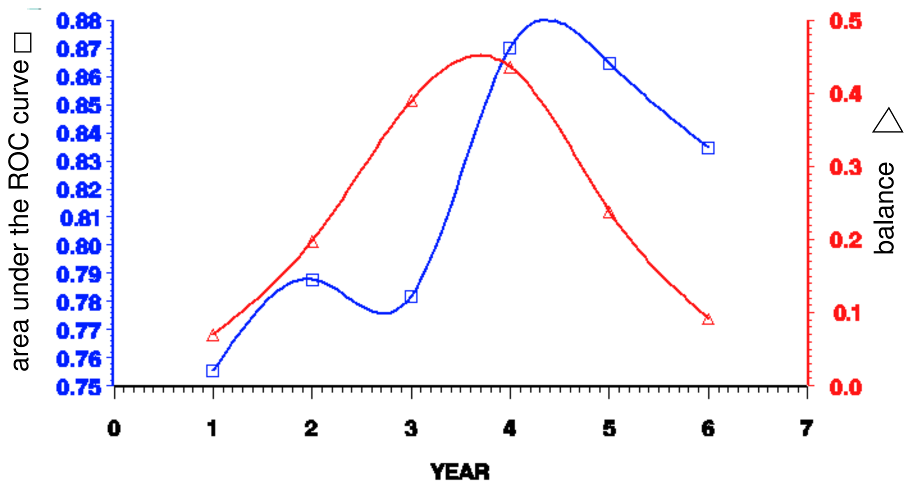
Table 8.2 shows the resolution of the models.

Table 8.7. Resolution of standard neural network models.

Year of follow-up	Area under the ROC curve	Standard Error
1	0.7554	0.032997
2	0.7879	0.021300
3	0.7818	0.197550
4	0.8703	0.017448
5	0.8647	0.020774
6	0.8346	0.031737

The best discrimination was achieved for a follow-up of four years. Figure 8.8 shows how resolution is related to data balance.

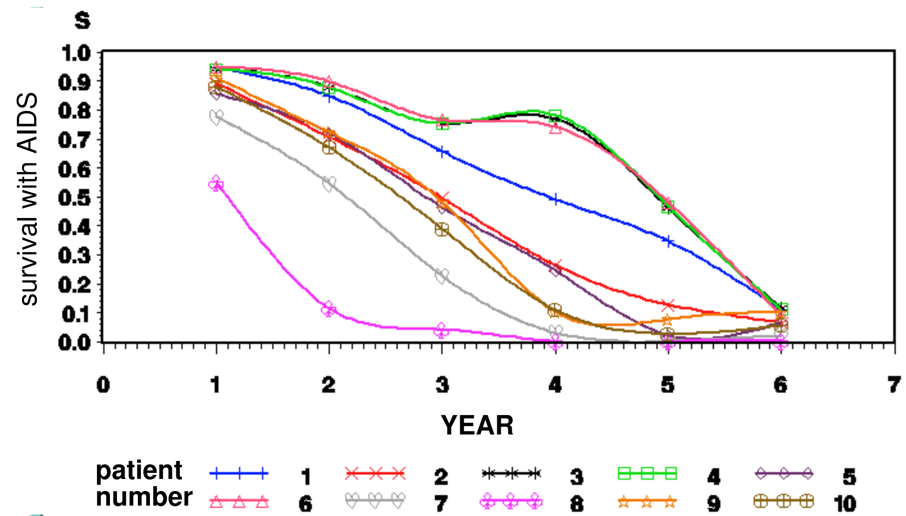
Figure 8.8. Resolution and data balance in standard neural network model.



Resolution is represented by squares and scaled in the left axis. Balance is represented by triangles and scaled in the right axis.

Figure 8.9 shows an example of 10 survival curves produced by the standard neural network models.

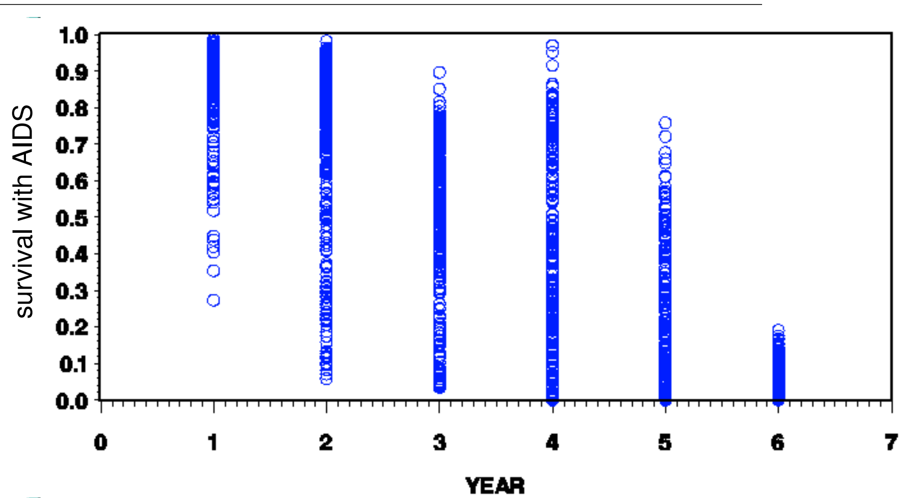
Figure 8.9. Survival curves for 10 patients using standard neural networks.



Since the models have no relation to each other, survival curves that are not monotonically decreasing, although impossible in theory, can be produced, such as for patient number 4.

The range of probabilities produced by the model is shown in Figure 8.10.

Figure 8.10. Range of probabilities for standard neural network model.



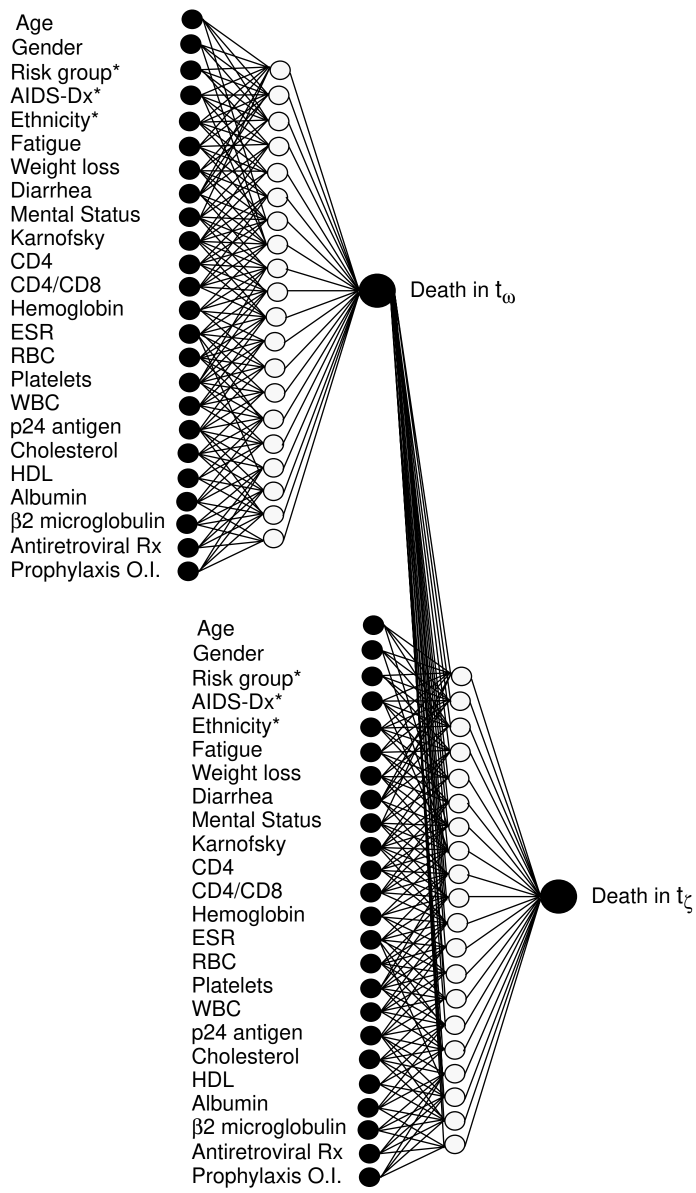
The range of probabilities for this model was wider than that for the Cox proportional hazards model, especially for intermediate intervals (years 3 and 4).

8.2.5 Sequential neural network model

A sequential neural network model, similar to the one shown in Figure 8.11, was built.

The neural network had 39 inputs (1 extra input for predictions in another year) and 20 hidden nodes. It was trained by backpropagation with adaptive learning rate. Overfitting was monitored in a holdout set of 40 percent of the cases. The software package NevProp2 [Goodman, 1994] was used.

Figure 8.11. Sequential neural network: ATHOS data.



In the sequential model, predictions for time t_ω are entered as inputs for the model that predicts death in t_ζ.

Table 8.12 shows the calibration of all sequential models.

Table 8.8. Calibration of sequential neural network models.

		Year of Prediction											
		1		2		3		4		5		6	
		χ^2	*p*	χ^2	*p*	χ^2	*p*	χ^2	*p*	χ^2	*p*	χ^2	*p*
Informative Year	1			8.8125	0.3583	21.677	0.0055	14.784	0.0634	28.754	0.0003	4.7113	0.7879
	2	10.933	0.2054			4.1572	0.8426	11.651	0.1674	25.116	0.0014	4.9896	0.7586
	3	10.661	0.2216	6.2774	0.6161			10.559	0.2279	28.646	0.0003	6.0991	0.6361
	4	7.6529	0.4680	9.2198	0.3241	25.431	0.0013			8.1632	0.4176	5.6195	0.6897
	5	10.364	0.2403	8.0985	0.4239	26.487	0.0008	11.274	0.1866			10.540	0.2291
	6	4.9829	0.7594	6.6582	0.5739	3.5031	0.8989	12.798	0.1189	33.434	0.0000		

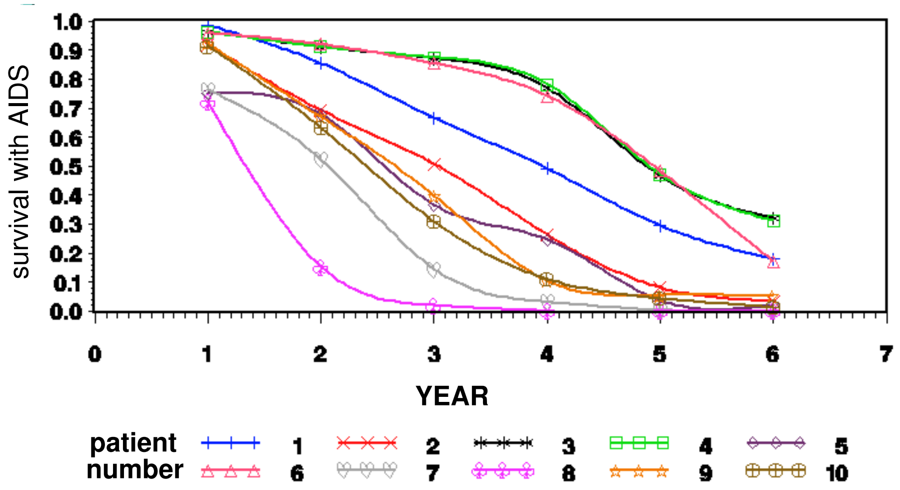
Table 8.12 shows the resolution of all sequential models.

Table 8.9. Resolution of sequential neural network models.

		Year of Prediction											
		1		2		3		4		5		6	
		Area under ROC	std. error	Area under ROC	std. error	Area under ROC	std. error	Area under ROC	std. error	Area under ROC	std. error	Area under ROC	std. error
Informative Year	1			0.7906	0.0210	0.7914	0.0194	0.8235	0.0191	0.8382	0.0224	0.8610	0.0324
	2	0.7888	0.0295			0.7973	0.0192	0.8234	0.0191	0.8367	0.0225	0.8646	0.0314
	3	0.8291	0.0223	0.7733	0.0226			0.8231	0.0192	0.8377	0.0224	0.8646	0.0329
	4	0.7912	0.0304	0.7936	0.0221	0.8063	0.0198			0.8720	0.0198	0.9065	0.0289
	5	0.8092	0.0261	0.7739	0.0227	0.7883	0.0199	0.8177	0.0196			0.8851	0.0301
	6	0.8008	0.0279	0.7890	0.0212	0.7986	0.0192	0.8222	0.0192	0.8285	0.0230		

Figure 8.12 shows an example of 10 survival curves produced by the sequential neural network models that have predictions of year 4 as inputs.

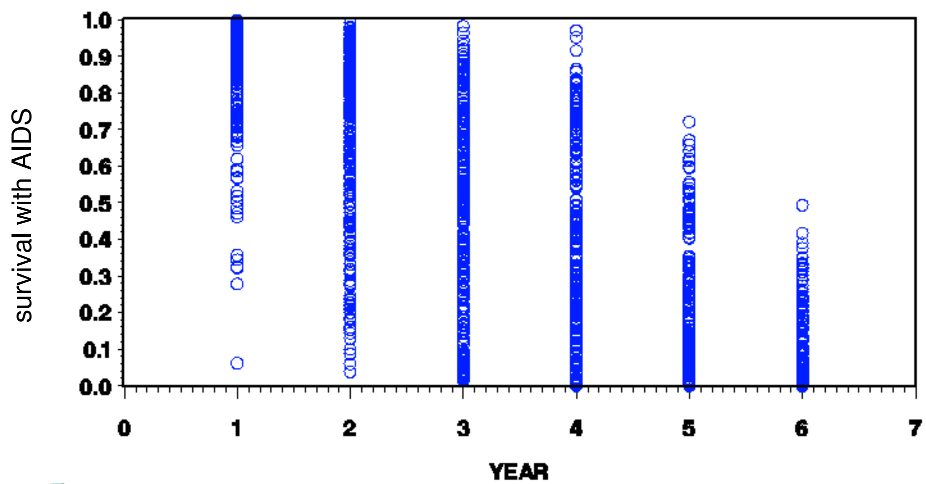
Figure 8.12. Survival curves using neural networks and information on Year 4.



Curves produced by the sequential models tended to have better spread and fewer nonmonotonic intervals.

The range of probabilities produced by the models that have predictions of year 4 as inputs is shown in Figure 8.13.

Figure 8.13. Range of probabilities using neural networks and information on Year 4.

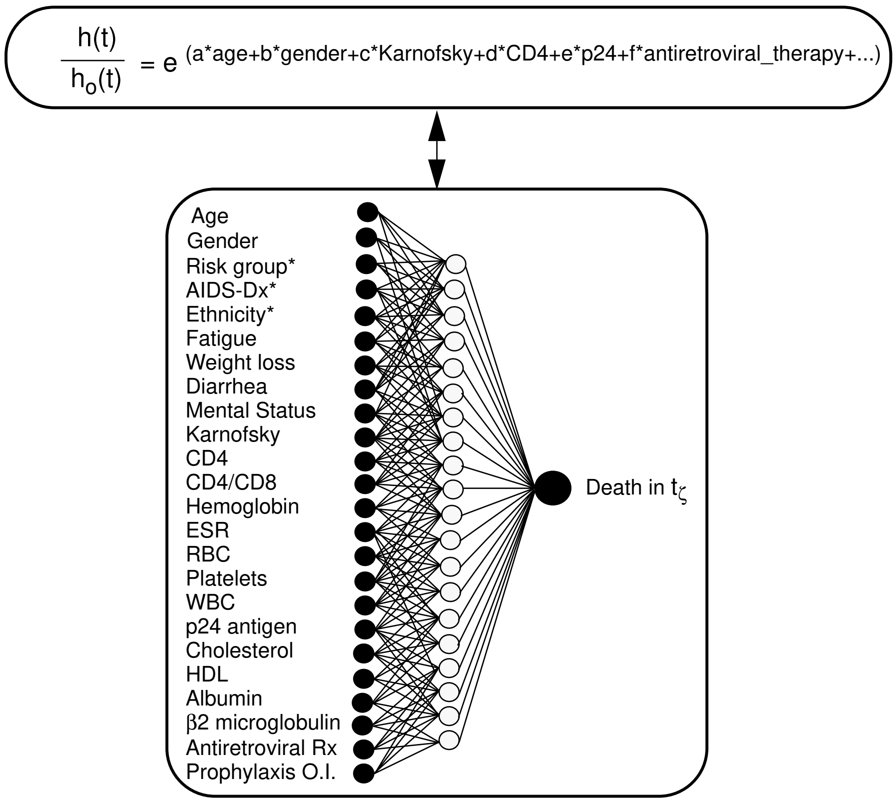


The range of probabilities for all intervals were higher than those produced by standard neural networks.

8.3 Model Comparison

Cox proportional hazards and standard neural network models, similar to the ones shown in Figure 8.14, were compared for calibration and resolution.

Figure 8.14. Cox proportional hazards versus standard neural network: ATHOS data.

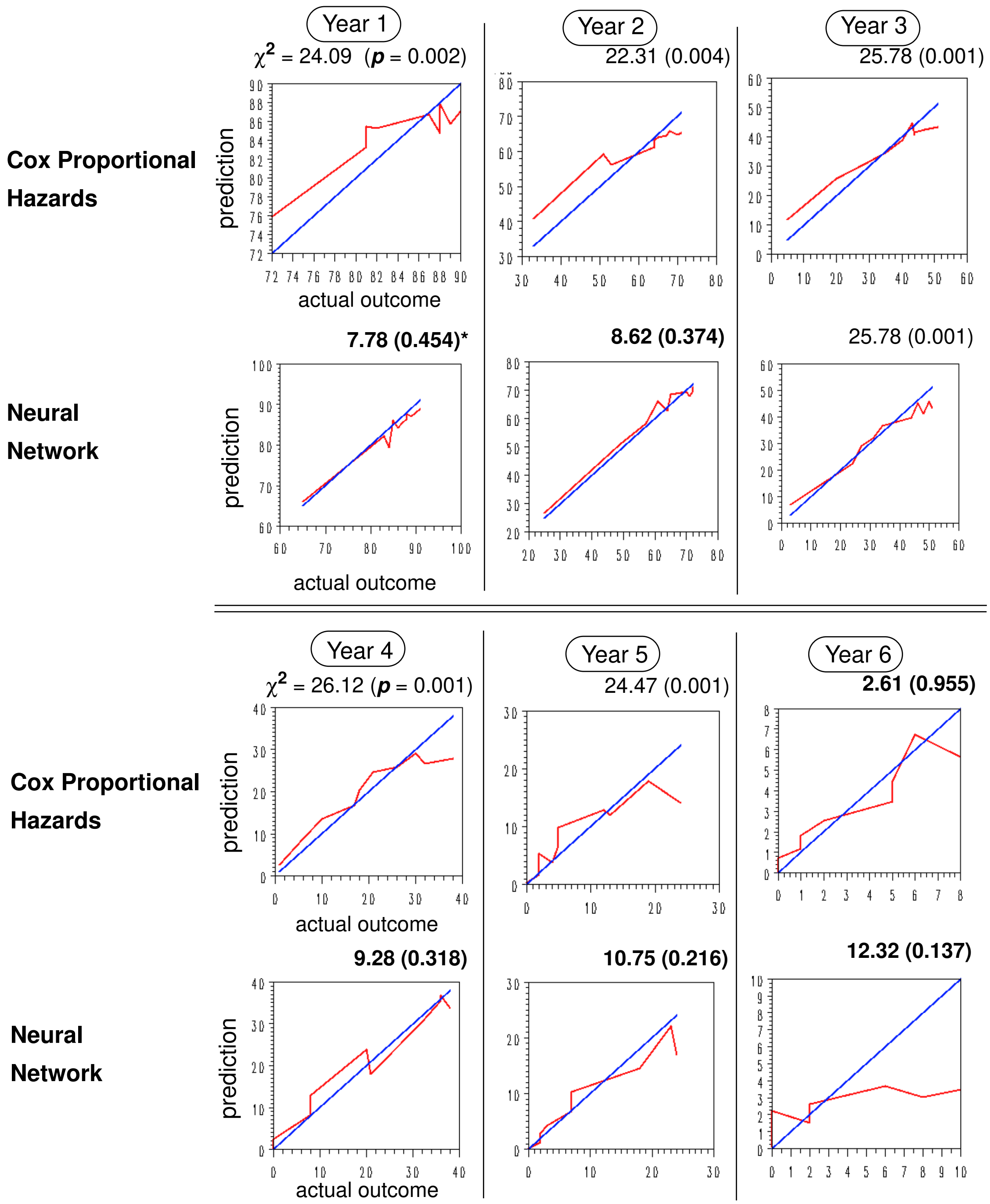


Training data provided to Cox proportional hazards models are more complete (i.e., Cox models are given the number of survival days for a given case, whereas standard neural networks are only given whether the patient was alive or dead at a given interval.

Cox models were given complete information about survival (e.g., 254 days), whereas standard neural networks were only given a binary assessment of survival (e.g., dead or alive at year 2). Comparisons were made for predictions at the end of each interval.

Figure 8.15 displays calibration plots for Cox and standard neural network models.

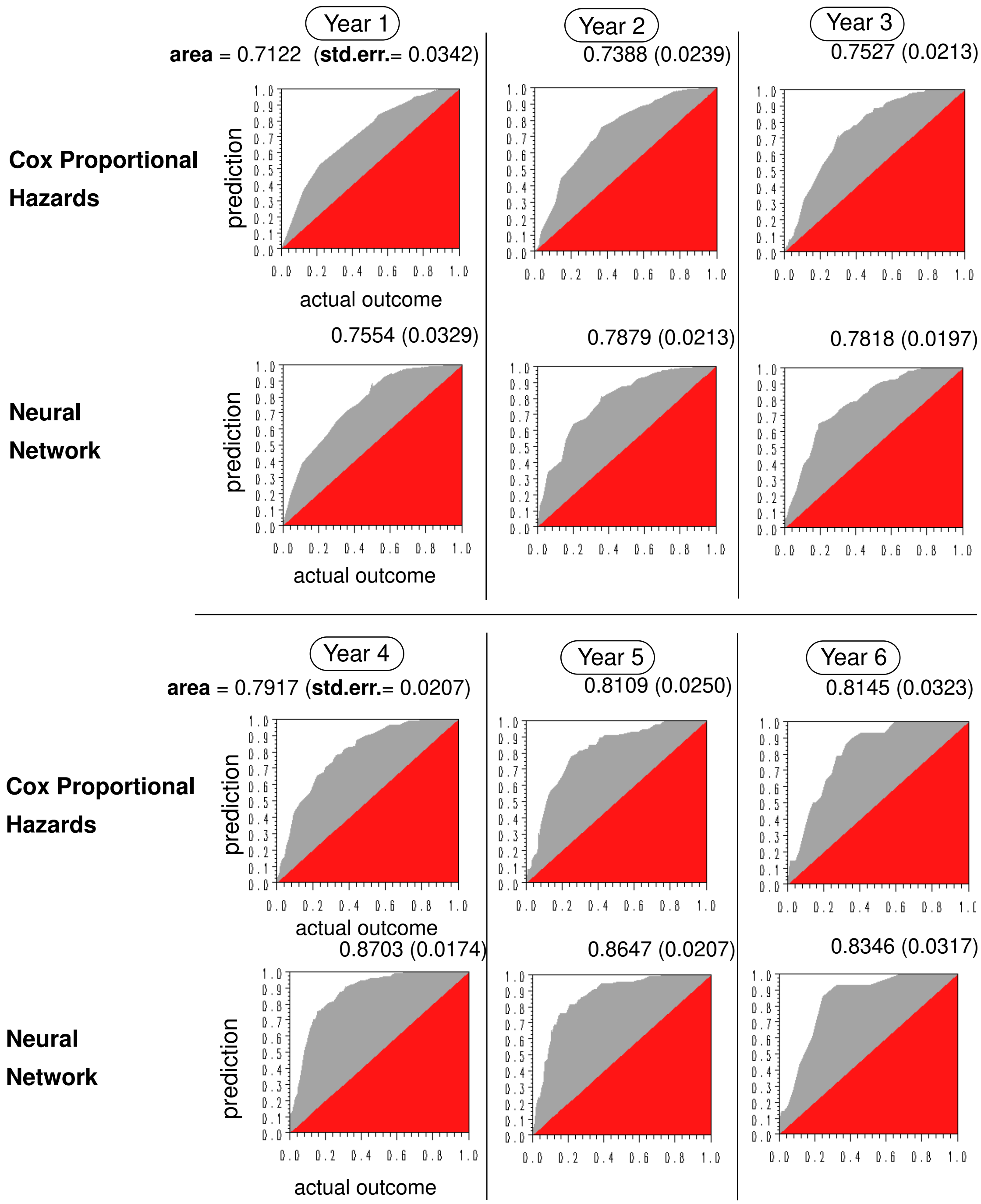
Figure 8.15. Calibration plots and Hosmer-Lemeshow χ^2 (*p*) for Cox and standard networks.



*Values in **bold** indicate good calibration. Note that axis scales differ.

Figure 8.16 shows areas under the ROC curves for Cox and standard network models.

Figure 8.16. ROC curves and areas (standard errors) for Cox and standard network models.



The darker triangles indicate no discrimination (area under the ROC curve=0.5).

The neural network model was better calibrated than the Cox model, except for predictions in the last interval (year 6). For that interval, neural network calibration was lower than that of the Cox model, but still considered good ($p = 0.137$).

Table 8.10 shows the differences in resolution between Cox and standard neural network models and their significance. Neural network models always provided larger areas under the ROC curve than did the Cox models. The differences were statistically significant ($\alpha = 0.10$) for years 2, 4, and 5.

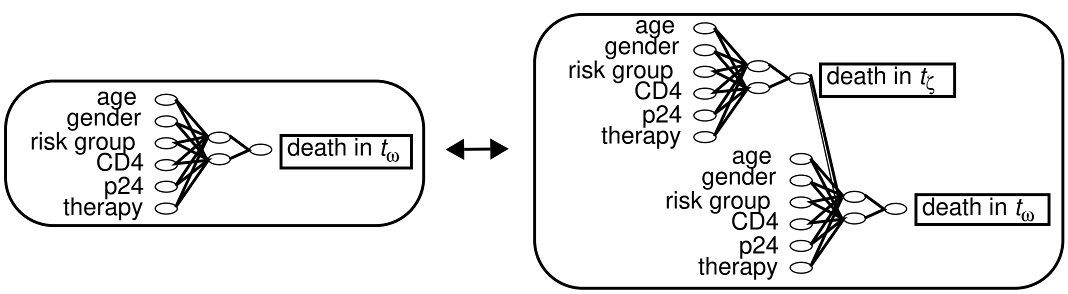
Table 8.10. Differences (*d*) in resolution between Cox and standard neural networks.

Year	*d*	*p*
1	-0.04320	0.17694
2	-0.04910	0.06089
3	-0.02915	0.15836
4	-0.07857	0.00186
5	-0.05381	0.04902
6	-0.02011	0.32861

8.4 Method Comparison

Standard and sequential neural network models, simplified in Figure 8.17, were compared for calibration and resolution.

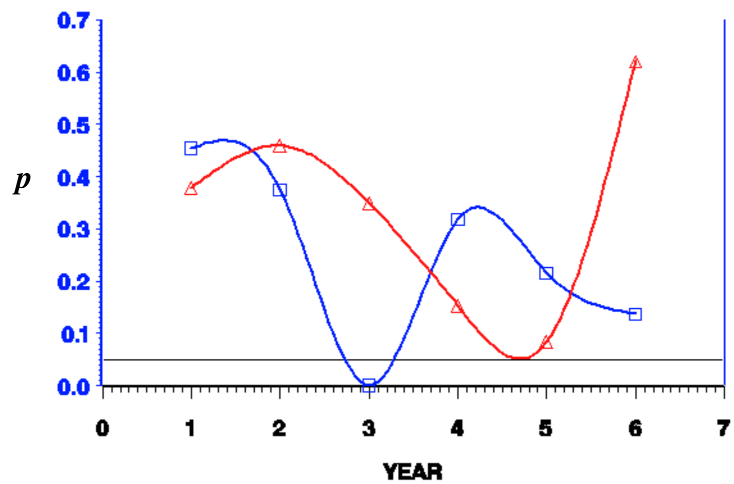
Figure 8.17. Standard versus sequential neural network: ATHOS data.*



Standard and sequential models were compared for calibration and resolution.
*Not all variables are shown in this figure.

Figure 8.18 shows the p obtained by the Hosmer-Lemeshow test on the standard models and the average p obtained by the same test on the sequential models. Overall, there was not a major change in calibration. Calibration plots (not shown here) did not show major departures from the expected numbers, indicating that calibration could be considered acceptable if the significance level of 0.05 were relaxed.

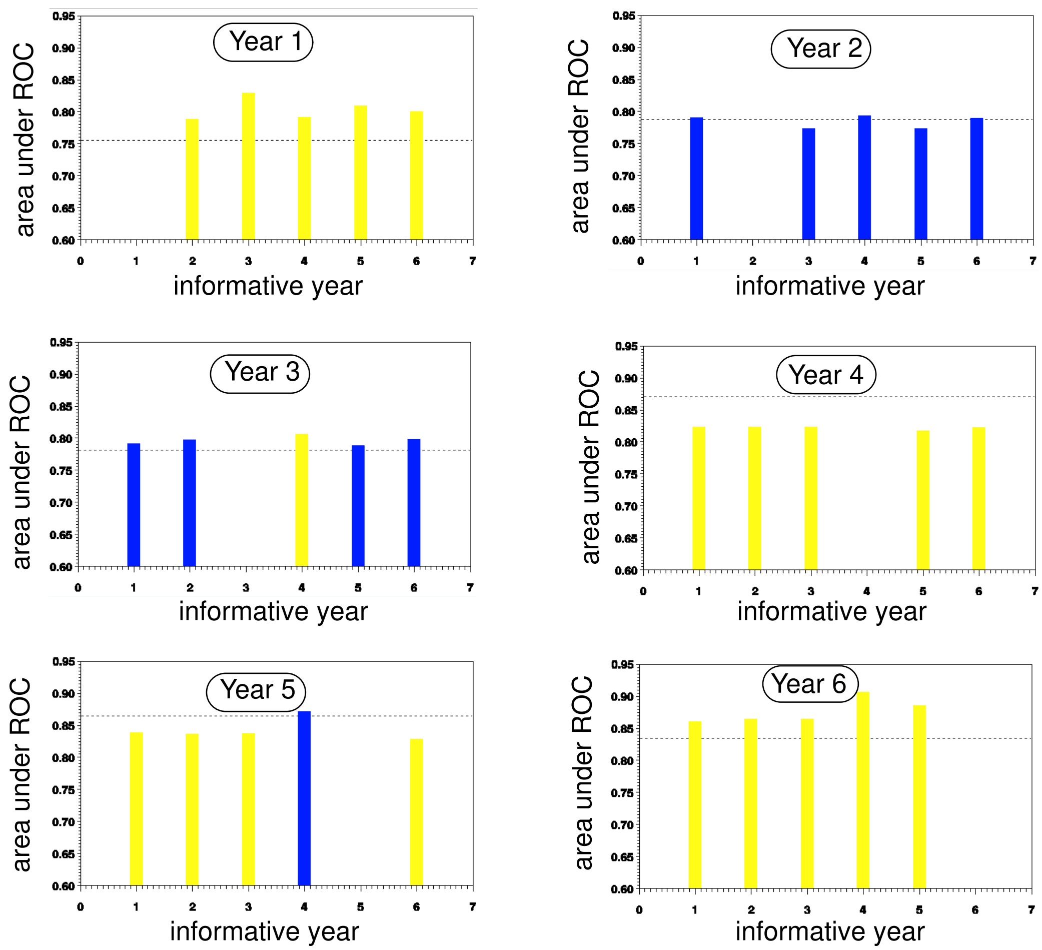
Figure 8.18. Calibration of standard and sequential neural network models.



The *p* (squares) and average *p* (triangles) obtained by the Hosmer-Lemeshow test are plotted for the standard and sequential models, respectively. The significance level of 0.05 is displayed for reference.

Of the 30 sequential neural networks, 11 resulted in significantly larger areas under the ROC curve ($p<0.10$) than their equivalent standard models, as shown in Figure 8.19. There was significant increase in resolution for years 1 and 6. The most informative year was 4. The most informed year was 1. The least informative year was 1. The least informed year was 4. Note that there was a significant decrease in resolution when information on other year predictions was added to the model that predicted death in year 4. The same type of decrease occurred for models predicting death at year 5, except when predictions for year 4 were provided, producing a sequential model with significantly higher resolution.

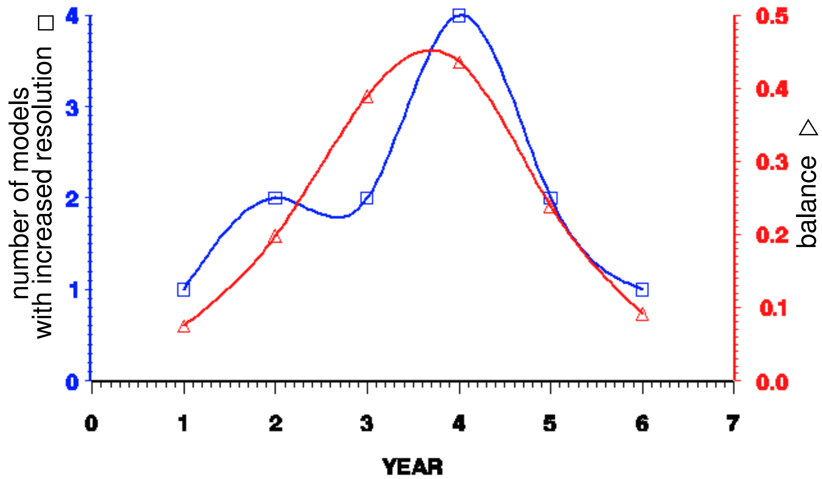
Figure 8.19. Resolution of sequential neural networks.



Bars represent the area under the ROC curve for sequential models when different informative years were entered in the model. The dotted line represents the resolution of the standard model for reference. Darker bars indicate that the difference in resolution was not significantly different for $\alpha = 0.10$.

Note, in Figure 8.20, that here the degree to which each year was informative is positively correlated with the balance of the data.

Figure 8.20. Informative years and balance in sequential neural network models.



Squares represent the number of times an informative year corresponded to a significant improvement in the area under the ROC curve and are scaled in the left axis. Triangles represent the data balance and are scaled in the right axis. The most informative years were the ones in which data was most balanced.

8.5 Summary of Results

I modeled disease progression for patients who are HIV+ in a large set of patients using Cox proportional hazards, standard, and sequential neural networks. The hypothesis that neural networks could make more accurate predictions of AIDS survival in terms of calibration and resolution than could Cox proportional hazards was confirmed. The hypothesis that sequential neural networks could make more accurate predictions of AIDS survival than could standard neural networks was confirmed. The intervals that were more informative were positively correlated with the balance of the data. It was not advantageous to use predictions for intervals where (a) data were not balanced or (b) the resolution in the standard model was poor. On average, there was not a significant difference in calibration between sequential and standard models.

Standard neural networks performed better than Cox proportional hazards in this data set. Sequential neural networks performed better than standard neural networks.

8.6 Discussion

The results reported here show performance on the whole subset chosen for this study. Overfitting in neural networks was controlled by monitoring the error in a holdout set of cases. Overfitting in the Cox model was limited by the fixed distribution function for the effects of the explanatory variables. However, controlling for overfitting does not mean that there is not underestimation of errors, since they are being calculated on a training set. The adjusted area under the ROC curve was calculated using the bootstrap method [Efron, 1983] for all neural network models. One hundred boots were used for each model, resulting in the adjusted areas under the ROC curves shown in Table 8.11 for standard neural network models. This areas are calculated on test sets that contain cases that were not used to build the models.

Table 8.11. Adjusted resolution for standard neural networks.*

Year of follow-up	Adjusted areas under the ROC curves
1	0.6871
2	0.7466
3	0.7415
4	0.8275
5	0.8106
6	0.7703

*Using bootstrap (100 boots for each model).

If a comparison is made between the *adjusted* areas under the ROC curves for the neural network models (Table 8.11) and the *unadjusted* areas under the ROC curves for the Cox proportional hazards models (Table 8.2), the differences still favor the neural network model. Table 8.12 shows the adjusted areas under the ROC curves for sequential neural network models.

Table 8.12. Adjusted error estimates for sequential neural networks.*

		Year of Prediction					
		1	2	3	4	5	6
		adjusted area	adjusted area	adjusted area	adjusted area	adjusted area	adjusted area
Informative Year	1		0.7471	0.7498	0.7756	0.7835	0.7909
	2	0.7286		0.7570	0.7774	0.7832	0.7954
	3	0.7661	0.7311		0.7772	0.7852	0.7972
	4	0.7351	0.7564	0.7692		0.8300	0.8784
	5	0.7476	0.7305	0.7466	0.7709		0.8492
	6	0.7389	0.7475	0.7578	0.7743	0.7725	

*Using bootstrap (100 boots for each model).

The fact that neural networks in general, and sequential neural networks in particular, provide better estimates of prognosis for this set of patients than does the Cox proportional model is not surprising. Neural networks are not limited by a parametric restriction on the relation between covariates and outcome. Complex nonlinear functions can be modeled by neural networks. This flexibility, however, is counterbalanced by the inability of neural network models to explain their results. As discussed previously in Chapter 2, neural network models cannot define which variables were the most influential in making a prediction.

CHAPTER 9

Discussion of Results and Conclusions

In Chapters 7 and 8, different experiments compared standard and sequential neural networks to statistical prognostic models in two medical domains, and discussed why certain results were obtained in those data sets. Overall, sequential models outperformed standard models in terms of resolution, without a decrease in calibration. Neural networks outperformed a Cox proportional hazards model in the ATHOS data set, but did not outperform logistic regression models using the Framingham data set. This chapter presents differences between and common aspects of the two experiments, and generalizes the results. Existing related work in hierarchical and sequential classification systems is also discussed. Section 9.1 describes the main differences in the experiments and results using the Framingham and the ATHOS data sets. Section 9.2 describes the similarities and generalizes the results. Section 9.3 explains the main steps involved in building an adequate sequential system of neural networks. Section 9.4 compares this work to other models for survival analysis.

9.1 Differences Between the Experiments: End-points and Data Collection

Two data sets were used to assess the performance of classical statistical models of survival analysis and standard and sequential neural networks. The Framingham data were used to develop logistic regression and neural network models that predict development of coronary heart disease (CHD), and the ATHOS data was used to develop Cox proportional hazards and neural network models that predict death due to AIDS. In the Framingham experiment, standard logistic regression and standard neural network models exhibited the same performance, as did sequential logistic regression and sequential neural network models. The performance of the sequential models was always better than that of their standard counterparts. In the ATHOS experiment, the standard neural network had better performance than the standard Cox proportional hazards model. The sequential neural network model had better performance than the standard neural network model. Although part of these results can be generalized, as explained in Section 9.2, some differences regarding the data collection, its utilization, and end-points of study deserve special attention.

The Framingham data set is not limited to the collection of data on people who already have an established disease, as is the ATHOS data set. For this reason, it can have many more cases than the ATHOS data set. Furthermore, the Framingham data collection is more extensive, since it started over 30 years ago, and it there is not a great number of missing data.

In the Framingham experiment, the inputs for the logistic and neural network models were values for a given patient at a specific exam, rather than at baseline, as in the ATHOS data set. In ATHOS, there was a specific entry criterion, AIDS diagnosis, so survival from AIDS to death was assessed. In the former, there was not a specific entry criterion (except the absence of CHD), and consequently no baseline values. It would be harder to interpret the results of a Cox proportional hazards model for this problem, since no absolute survival from a baseline date for a given patient (e.g., 234 days) could be provided, but only survival relative to the date of a specific exam (e.g., 234 days from the 2/1/84 exam). In this work, survival without CHD was assessed using logistic regression models and neural networks. For both models, standard and sequential versions could be built.

In the ATHOS experiment, a standard Cox proportional hazards model was built and survival predictions were assessed for years 1 through 6. Because no information can be added by simply including a variable representing predictions in other time points in a Cox model (the absolute survival in days is already given for a certain patient and it drives the construction of the whole survival curve for that patient), a sequential Cox proportional hazards model was not constructed. Both standard and sequential neural networks were built. Since the comparison between standard neural networks and the Cox proportional hazards models resulted in improved performance for the standard neural network, and the comparison of the standard and sequential neural network models resulted in improved performance for the later, a direct comparison between the sequential neural network and the Cox proportional hazards model was not necessary.

In the Framingham experiment, the predictions of either standard or sequential neural network models were not significantly different from those of the corresponding logistic regression models. In the ATHOS data set, both the standard and the sequential neural network models had higher calibration and resolution than those of the Cox proportional hazards model. Evidently, it is not true that neural network models always produce better estimates than equivalent statistical models. Different models make different assumptions about the distribution of data, such as hazards proportionality. The assumptions made by the logistic regression models with respect to the Framingham data were not as restrictive and probably more adequate than the ones made by the Cox models with respect to the ATHOS data. For researchers dealing with the same task (i.e., prediction of survival with AIDS and prediction of CHD development) and the same subsets of data from the ATHOS and the Framingham data set, the results of these experiments can serve as benchmarks for comparison of predictive performance for different models.

Although there were differences between the experiments with the Framingham and the ATHOS data sets, the main results regarding the hypothesis that sequential neural networks perform better than standard neural networks were the same. The following section describes the results that were common to both experiments.

9.2 Common Results Using the Framingham and the ATHOS Data Sets

The portion of this research that compares the performance of standard and sequential models is the one that can be generalized to other tasks and other data sets. In both experiments, sequential models had better resolution than standard models, without sacrifice of calibration. The reason for this improvement was simply the addition of a standard model's prediction for another time point. For example, predictions obtained from a standard neural network that modeled death within four years in the ATHOS data set were added to the model that predicts death within one year. What the standard models lack are exactly the dependencies that need to exist between predictions for different intervals. The dependency can exist both ways: If knowing the predictions for year 4 helps to make better predictions for year 1, the reverse will be true (but only if both the balance of the data in the informative year and the resolution achieved by its standard model are higher than those of the informed year, as indicated by the result of both experiments).

In order to understand why certain intervals yield models with higher predictive ability than others, let us first explore a simple example. Let us then see why the intervals that would apparently be the easiest ones to predict may be the ones that pose the greatest challenges, due to the problem of recognizing infrequent examples. Suppose that I wanted to estimate the probability of survival for all my friends who are currently in their late twenties and early thirties (excuse me for the morbid example) for several points in time: 1 year, 40 years, and 70 years from today. Which time point could I predict with highest accuracy? Seventy years from now we will probably all be dead, so that prediction seems easy: 0 percent for all. One year from now we will probably all be alive, so prediction seems easy for all: 100 percent for all. Note that this type of prediction is not discriminating at all, resulting in areas under the ROC curves that are very close to 0.5. Forty years from now, however, some of us will be alive and others will be dead. Making an estimate based on personality types, cigarette smoking, cholesterol levels, exercise levels, etc., given our previous experience, can yield models that are more discriminatory, because data are more balanced at this time point (forty years) than at the extreme time points (one and forty years). The area under the ROC curve would be larger than 0.5. Making the same type of estimate for 1 and 70 years is not as easy.

In a sequential model, I can use the predictions for 40 years (since it is the most discriminatory) as a starting point to minimize the problem of recognizing infrequent patterns in 1 or 70 years. If there is a high probability that a person will be dead in forty years, that person will probably be dead in 70 years. Conversely, if I want to know the probability of survival in one year, I can work backward: If a person had a high probability of being alive in 40 years, he probably has a high probability of being alive in 1 year. Since survival has to be a monotonically decreasing function, it would seem intuitive that making accurate predictions for any time point would facilitate predictions for other time points, just as described above. However, the results in both data sets show that predictions for time ζ, when used as inputs for sequential models that predict survival in time ω, are only useful when (a) the predictions for time ζ were made with high accuracy (at least higher accuracy than those of the standard model for time ω) and (b) the balance of data in time ζ is higher than that in time ω. These are necessary conditions for a significant improvement to be detected, as the results have shown.

In the next section, I use these conclusions to develop guidelines on how to build sequential models. Then I examine, in Section 9.4, how some current models of survival analysis and time-series forecasting relate to this research.

9.3 Lessons Learned: How to Build Sequential Neural Networks

As we saw in the experiments described in this dissertation, not all sequential implementations of neural networks or logistic regression models result in better predictive models. Certain constructions improve the resolution provided by standard models, whereas others can bias the predictions. Possible reasons for this behavior have been discussed in Chapter 9. Overall, the results indicated that the following steps should be taken when constructing a two-step sequential system:

1. Start by building and assessing calibration and resolution in standard models for all outcomes. In the case of prognostic systems related to survival, the outcomes are survival predictions for various intervals of time. In diagnostic systems, outcomes are the most specific, detailed, diagnoses.
2. If the standard models are deemed adequate and there is no justification for spending more time and resources in building sequential models, stop. Otherwise, proceed to step 3.
3. Build supersets of outcomes with common characteristics, using data that is well balanced. In the case of survival analysis, supersets are natural: the set of patients alive at year *n* belongs to the superset of patients alive at year *n-1*, and so on. Conversely, the set of patients dead at year *n* belongs to the superset of patients who are dead at year *n+1*. Two supersets can then be defined using the year with the most balanced data as a separator. For example, if the balance between dead and alive at year 4 is the best, build two supersets of patients: one for those alive at year 4 and another for those dead at the end of that time interval.
4. If the resolution of the model that discriminates data in both supersets is not greater than that of the standard model that tries to identify a given outcome, then stop and try other separators. Otherwise, produce a

prediction for the separator outcome (e.g., year 4) and apply the predictions to another model (e.g., a model that predicts death in one year).

The generalization to a three-or-more-step sequential system is an extension of these guidelines.

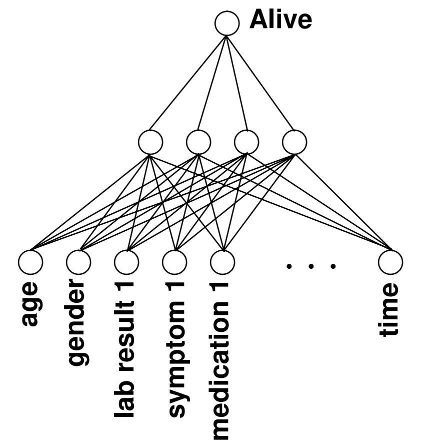
9.4 Relations to Other Prognostic Models

9.4.1 Relation to other neural networks models for prognosis

As discussed previously in Chapter 5, the intention of the work described in this dissertation was to show that neural networks can be used to build individualized prognostic curves for a given patient (and not only single point estimate of survival) even when data are unbalanced. Building a prognostic curve allows a temporal pattern of disease development to be delineated and makes possible the recognition of abnormal patterns or variations within normal patterns.

Many of the existing prognostic neural network models are intended to provide a specific point estimate of survival (e.g., survival in five years) or a continuous estimate of survival (e.g., 432 days). By contrast, the work of Ravdin [1992], where time is considered an input variable and variable values for individuals in different time points are provided as inputs, is an exception. It allows the construction of survival curves if necessary. The architecture can be seen in Figure 9.1. It has the advantage of being very simple, since a single network models predictions for all intervals. It involves considerable preprocessing of data, with selective duplication of cases that have longer survival. For example, if patient A lives five years and patient B lives one year, the input data set may contain five copies of patient A (one for year 1, one for year 2, etc.), but only one copy of patient B. Ravdin has developed a method to account for this bias in the final prediction. This architecture uses a single function to map inputs to outputs (patient features to predictions of survival), and it may be a complex function if the number of hidden nodes is large.

Figure 9.1. Example of Ravdin's architecture for survival analysis.



Time is part of the input in this architecture. Values for the same patient at different times constitute the training set.

The sequential system described in this dissertation used multiple functions to map inputs and outputs. From the results of Chapters 7 and 8, we can see that there are significant differences in calibration and resolution for models that predict outcomes in different time points. For example, the resolution of a model that predicts death in four years for a patient with AIDS is higher than that of a model that predicts death in six years. If we establish a threshold on calibration and resolution, we may determine which standard models provide "reasonable" performance and refrain from making predictions for other time intervals in a nonsequential way. To make those predictions, we use the predictions of "good" standard models as inputs to our sequential models.

9.4.2 Relation to ARIMA models for time-series forecasting

The sequential models used in the Framingham experiment described in Chapter 7 resemble those used for time-series forecasting, especially time-series regression [Bowerman, 1987]. This special type of regression combines Box-Jenkins methodology (also called Autoregressive Integrated Moving Average—ARIMA models) with regression analysis. The continuous outputs in time-series regression for time *t* are dependent on a number of variables, including the outputs for time *t-1*. For example, the probability of survival without CHD at year 2 is dependent on a patient's values for age, cholesterol, etc., and also on the probability of survival at year 1.

There are important differences between the sequential models used in the experiments described in Chapter 7 and time-series regression models. First, the outputs in the former did not need to be equally spaced in time. Second, the sequential models were built in either ascending or descending order with respect to time. We have seen that the sequential models that turned out to be the most useful were often constructed in descending order: using predictions for year 20 as inputs for the model that predicts CHD in year 10 was more useful than the reverse. In time-series regression, a trend is usually modeled only in ascending order. Third, time-series regression uses information on the whole series to build a curve that fits the historical data and to extrapolate and make new predictions. The sequential models presented in this work only used one other time point prediction as input (although using more time points could have also been done, as will be discussed in Chapter 10.

Comparisons between neural networks and other classification models are not new in the literature. Depending on the assumptions required and how they are verified in the available data sets, it is possible to show that certain models have better performance than others. In this research, I have shown that neural networks are superior to Cox proportional hazards models for prognosis of AIDS patients using the ATHOS data set, whereas neural networks were not superior to logistic regression models for prognosis of CHD development using the Framingham data set. The results of these experiments illustrate how important it is to start modeling with the simplest and most interpretable models, and then to assess possible improvements by applying novel techniques. These results cannot be generalized to other tasks and other data sets, but can serve as benchmarks for other researchers working with the same data sets.

When comparing the standard and sequential models, however, I showed that in both the Framingham and the ATHOS data sets, sequential models improved resolution over standard ones, and I provided an explanation for this result, indicating that that will always hold under certain circumstances, regardless of the tasks or the data sets involved. These results can be used by other researchers who need to model prognostic tasks over time.

Even though most of the results shown in Chapters 7 and 8 indicate good prognostic performance for the various models, the information provided by the absolute areas under the ROC and the definition of what constitutes a significant difference from the point of view of the health care worker and the patient depends on a variety of other factors. Trade-offs between predictive performance and resource utilization by the several models have not been addressed here. Healthcare provider and patient utilities were not taken into account, either.

CHAPTER 10

Summary and Future Work

The anticipation of events is an essential part of the practice of medicine. Important decisions regarding the pursuit of aggressive diagnostic or therapeutic interventions are based on the balance of expected costs and benefits. Current data on patient outcomes make it possible to develop models to predict development of disease using multiple variables. These models can exhibit better performance if developed sequentially, in such a way that most information is utilized. I developed and tested a sequential model of neural networks that allows accurate prediction of disease development over time. This model can help health care providers and patients anticipate events with more precision and therefore make more informed decisions.

The preceding chapters have presented and discussed (1) the problems related to the recognition of rare categories in machine-learning methods, (2) existing deficiencies in current prognostic models and the advantages of using neural networks, and (3) the results of experiments that demonstrate that sequential neural networks provide predictions that are associated with high resolution and calibration.

This chapter provides a summary of this dissertation. Section 10.1 discusses the significance of the problem of recognizing rare categories in medicine, and the impact of this problem on the performance of models that predict outcomes. Section 10.2 presents sequential neural networks as a solution to this problem. Section 10.3 restates the main hypotheses of this research and highlights the results and conclusions derived from two different experiments. Section 10.4 discusses possible extensions to this work. Section 10.5 provides an overview of the contributions of this work to the fields of medicine and information sciences.

10.1 Significance

If the practice of medicine were limited to diagnosis and treatment of common illnesses, there would be little place for sophisticated learning and specialization to take place: a simple memorization of adequate protocols of assessment and interventions would suffice. It is the existence of rare conditions and the need for individualized assessment and treatment that make the practice of medicine challenging, and that help to distinguish those who simply practice “cookbook” medicine from those who master the “art of medicine.” To a certain extent, the same is true of machine-learning models. While the performance of several machine-learning models has been shown to be good on average (especially in the recognition of common conditions), it is often the ability to recognize infrequent patterns and to differentiate certain patterns of disease that differentiates good and bad models. Current machine-learning models of diagnosis in medicine, including neural networks, do not easily or accurately recognize rare categories or discriminate patterns with sufficient precision [Lowe, 1990]. The same can be said for current machine-learning models for prognosis. Current solutions are inadequate.

The prognostic assessment of disease progression is an essential part of medicine. From the caregiver’s perspective, predicting outcomes for a given patient influences therapeutic decisions. From the policymaker’s perspective, predicting outcomes for a subset of a population influences allocation of resources. From the patient’s perspective, predicting outcomes influences many aspects of life: financial, professional, and emotional. The availability of electronic databases and new computer-based technologies has made it easier for the practice of evidence-based medicine to be extended to the prognostic assessment of disease progression: statistical models for prognosis of patients in special settings (e.g., ICU), or with special conditions (e.g., trauma) have received increased attention from all those involved in the health care industry. As these methodologies evolved, methods that once illustrated principles and were restricted to the academic community as recently as ten years ago can now be used at the office or the bedside. Making a prognosis for an individual patient based not only on previous experience, but also on the experience of others—gathered from the literature and analyzed by quantitative methods that require the use of a computer—should be the rule, rather than the exception, in the practice of contemporary medicine.

Neural networks have a relatively short history of utilization in medicine, and their potential in the field is not yet fully understood. As with any other new method, neural networks have suffered the criticism of people who did not understand them (but who were still eager to point out their deficiencies). Neural networks' need for large amounts of data, their slowness to estimate parameters, and their inability to explain the relative importance of variables are some of the complaints of those who still believe these "black boxes" will never have a place in the gallery of well-accepted statistical methods, such as linear and logistic regression, and Cox proportional hazards (in survival analysis applications), to name just a few.

The requirements for neural networks are approximately the same as those for regression models, especially when there are many variables involved and potential interactions among terms need to be taken into account. No neural network researcher has ever advocated the use of neural networks for simple univariate problems, which can be easily solved with current regression models. But medical problems can seldom be so simplified. Neural networks are usually applied to problems in which the classification has to be done with respect to multiple dimensions, and for which no simple causal relationship can be derived. This multidimensionality not only implies the use of several exemplar cases to develop a neural network model, but also determines the difficulties in the interpretation of its results. Yet, although it is desirable that a model indicates causal relations between independent and dependent variables, we cannot discard the importance of neural networks as predictive tools. The medical community, as opposed to the financial or engineering community, has been reluctant to accept a method that admittedly (1) is not guaranteed always to provide the best solution and (2) cannot be easily interpreted, even though it has been shown to provide accurate predictions. If the purpose of the model is explanatory, then that might be a reason to support this attitude. If the purpose of the model is to provide accurate forecasts, then there is no reason at all not to use neural networks in medical applications.

The prediction of outcomes for an individual patient is dependent on several variables. Unknown interactions, as well as noise, may influence the results. Although neural networks have been shown to be resilient to noise and able to handle interactions, their predictive accuracy is severely limited when the data are not well balanced (i.e., the priors for some outcome classes are low). This limitation is not exclusive to neural networks, and current methods for decreasing its impact on classification accuracy have been applied to other classification systems as well: equalization of priors (by sampling the training set in a way that would make the representation of classes more balanced) or application of cost functions (or utility functions) in parameter estimation. The problems with these two approaches were discussed in Chapter 4.

The assessment of prognosis for patients over time illustrates the need for dealing with the problem of unbalanced data: at the extremes of the time intervals that represent the duration of a disease or the life span of a human being, there are often time points in which the data represent few people with or without a certain condition (e.g., dead, in the case of the initial time points in a study of survival). In these cases, the classification of infrequent exemplars is hard. The sequential application of neural networks to partial subtasks facilitates the recognition of these infrequent cases, without an impairment of total classification accuracy. I have shown in this dissertation that sequential application of neural networks or logistic regression models to the prognosis of patients provides results that not only are more accurate in terms of discrimination (especially for infrequent cases) but are also more realistic, since they incorporate the commonsense knowledge that predictions of survival are necessarily correlated over time. Sequential methods make more use of the available information, and can significantly enhance the predictive ability of current models of prognostic survival analysis, delineating patterns of disease progression that could not be envisioned by current methods. This increase in predictive ability will (1) empower patients, since they will have more precise estimates as to how their disease will progress, (2) empower health care givers, who will be able to make more informed decisions on the course of therapeutics, and (3) empower health care organizations, which will be able to anticipate the needs of their covered population and anticipate costs.

Survival analysis can be viewed as a problem in which rare categories of events need to be discriminated. Standard neural networks can be accurate predictors, provided that the frequency of events is not low. Sequential neural networks provide a way to achieve high accuracy even for low-frequency events. I applied sequential neural networks to two medical problems, and compared their performance in terms of calibration and resolution to that of more conventional statistical models. The Framingham data set was used to study coronary heart disease development and the ATHOS data set was used to study survival with AIDS.

The Framingham and ATHOS data sets, which I used to illustrate the problem of recognizing infrequent outcomes and the improvement in accuracy achieved by a sequential neural network model, each addressed a different, though extremely important, domain in the medical field.

Coronary Heart Disease (CHD) has a high prevalence in developed countries, and is responsible for the majority of deaths in adults. The understanding of factors that influence the development of coronary disease continues to be a challenge for health care researchers. Currently, logistic regression models are the most frequently used in survival analysis in this domain. I have shown that neural networks can provide good models to predict death from CHD, especially if built sequentially.

The ongoing AIDS epidemic has posed new challenges for disease modeling. Not only are the relevant variables for predicting death not fully validated, but also the disease, having emerged only a little more than a decade ago, has a short history of follow-up. Accurate models of survival analysis for AIDS patients can be useful either on a one-to-one basis for advising a patient, or on a large-population scale for developing health care policies. Currently, the most frequently used model of survival analysis in the AIDS domain is the Cox proportional hazards model. The performance of this model is dependent on assumptions that have been shown to be not always satisfied with actual data from AIDS patients. Neural network models, in general, provide a good alternative for modeling AIDS survival, and sequential neural networks, in particular, can provide accurate predictions of death due to AIDS.

In both the Framingham and the ATHOS data sets, there were intervals of time for which data were unbalanced. For both data sets, it was important to accurately predict survival in those intervals in order to delineate an individualized survival prognostic curve for a given patient.

10.2 Sequential Neural Networks

A sequential system of neural networks was presented as a solution to the problem of recognizing infrequent patterns in survival data. The sequential system makes use of accurate predictions for a certain time point to develop a model that makes predictions for other time points in which the accuracy is not as high. The use of this type of information not only allows an increase in resolution for certain time points, but also increases the model's overall consistency, by producing survival curves that have fewer nonmonotonic intervals. The sequential system does not require that the predictions are made in ascending order. As we learned in the experiments described in Chapters 7 and 8, the important features for a specific time-point prediction to be used first in a sequential model are that (1) data are balanced in that time point and (2) the resolution of predictions is high.

Sequential neural networks are easy to build, and do not require any change in the learning algorithm. Certain sequential models may, however, require longer training times than their standard counterparts. In the example of Chapter 7, sequential logistic regression models outperformed standard logistic regression models.

10.3 Hypotheses and Overall Results

I tested the hypotheses that (1) sequential neural networks produce results that are more sensitive and more specific to infrequent patterns than nonhierarchical neural networks, given shorter training times, and (2) in certain circumstances, neural networks produce better estimates of survival time than (a) logistic regression models or (b) Cox proportional hazards models.

My first hypothesis was tested in both the Framingham and the ATHOS data sets. In both experiments, sequential neural networks exhibited better performance in terms of resolution than the standard networks, so that hypothesis was accepted. In the Framingham data set, I further tested whether a sequential logistic regression system performed better than standard logistic regression, again obtaining significant improvements in resolution, with no sacrifice of calibration. The results indicated that sequential models were more accurate than their equivalent standard models, regardless of whether they were based on neural networks or on logistic regression.

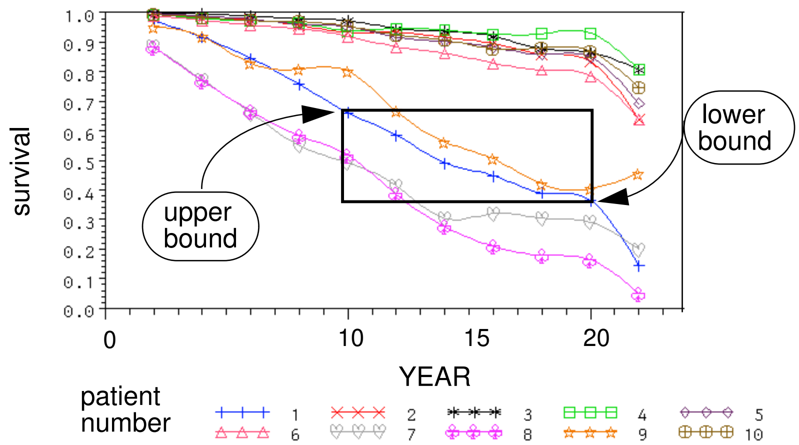
Hypothesis (2a) was tested in the Framingham data set and was rejected. In that data set, there were no significant differences between the performances of logistic regression models and neural network models. The relation between covariates and outcomes was well fitted with the logistic function, and no improvement could be verified when neural networks were used. However, since neural networks can model a large set of functions, and their performance is potentially at least as good as logistic regression, their use can be justified. The discussion in Chapter 7 also provides some clues as to why the performance of neural networks was not better in this experiment.

Hypothesis (2b) was tested in the ATHOS data set and was accepted. Neural networks had better predictive performance than Cox proportional hazards models in this data set.

10.4 Future Work

Only two-step sequential systems were described in this dissertation. The use of more steps would imply using more computer resources, but would have the potential to improve still further the resolution of certain models. For applications that try to determine survival, for example, a three-step model that first predicts the extremes of an interval and then applies those predictions to the model that predicts the middle of the interval may improve resolution. Suppose that the researcher is only interested in the 15-year survival for a certain group of patients. If the data are such that predictions for survival in 10 and 20 years using a standard model are more accurate than those for survival in 5 years using the same kind of model, then the predictions for years 10 and 20 could be used as inputs to a sequential model that predicts survival in 15 years. The improvement would result from establishing accurate bounds to the range of probabilities that are produced by the model that predicts survival in 15 years, as shown in Figure 10.1.

Figure 10.1. Bounding the range of probabilities in a sequential model.



Predictions of a sequential model in the middle of an interval can be more accurate if they are bounded by predictions for the extremes of the interval, usually produced by standard models. In this example, predictions for year 15 in a sequential model would fall in the interval 0.33 to 0.68.

Other extensions to this work involve the comparison of neural networks and other nonparametric models, such as regression trees, and validation in other data sets and other domains.

10.5 Contributions

I expect that the results of this work will encourage more widespread use of neural networks in certain types of medical applications, and that this use of neural networks will sometimes produce models that are more accurate than currently used statistical models. The main contribution of this work is conceptual: I did not create backpropagation, I simply demonstrated how this popular algorithm can provide better results in certain medical problems if applied in a certain manner.

10.5.1 Contribution to medicine

Currently used backpropagation-based neural network models to classify medical patterns, or to forecast medical events, generally have difficulty learning infrequent patterns. By showing that there is a significant improvement in resolution with sequential neural networks, without a decrease in calibration, I broadened the spectrum of medical applications that can benefit from neural network models. In particular, I demonstrated how sequential neural networks can be used in survival analysis for predictive purposes, providing accurate results without a need for assumptions usually required by conventional statistical methods. I have also shown that other models of prediction, such as logistic regression models, can provide accurate results if utilized in a sequential architecture.

The researcher confronted with the problem of forecasting events or establishing projections of survival curves for individuals, including for intervals of time in which data are unbalanced, should benefit from the sequential utilization of prediction models described in this dissertation, especially neural network models. As I showed, the predictive performance of neural network models is at least as good as that of other models, and sequential models have the ability to increase discriminatory performance for intervals in which prediction accuracy is poor. Clearly, simple models of prediction should always be tried first. These models are generally more economical in terms of computer resources, and generally provide results that are easy to interpret. The assumptions required by some of these models are, however, often unrealistic. Neural networks can be added to the gallery of tools available for the epidemiologist who wants to make predictions for a population, for the physician who is confronted with an individual case, and for the patient who wants to know more about his or her condition.

10.5.2 Contribution to information sciences

The main contributions to information sciences are (1) to demonstrate that backpropagation-based neural networks can be used even when the frequency of certain events is low, by using a hierarchical system of networks (or its generalization, a sequential system of neural networks) and (2) to describe the learning enhancement of certain patterns in backpropagation-based neural networks when additional structural information is added to the model (e.g., intermediate grouping abstractions). A method that utilizes sound principles for qualitatively modeling classification tasks into useful hierarchies, and that establishes some requirements for building sequential neural networks, is also a significant contribution of this work to information sciences.

There is still a long way to go to reach full recognition of neural networks as acceptable models of disease progression. The initial prejudice against connectionist models is slowly being eroded by increasing interest in these models, bolstered by their undeniable success in other disciplines and by rigorous evaluation. It is my expectation, therefore, that the contributions of sequential neural networks to survival prediction will be recognized first by the nonmedical community. In other domains, where the researchers admit their lack of ability to establish causal relations, and where the prediction of events is sometimes more important than their explanation (e.g., prediction of earthquakes, prediction of stock market behavior), sequential neural networks may be easily adopted. Acceptance by the medical community will depend, among other things, on their success in other domains.

REFERENCES

Akaike H. Information theory and extension to the maximum likelihood principle. In Petrov BN and Czaki F (eds), *Proceedings of the 2nd International Symposium on Information*. Akademiai Kaido, Budapest, 1977.

Akay M; Welkowitz W. Acoustical detection of coronary occlusions using neural networks. *Journal of Biomedical Engineering*, 1993 Nov, 15(6):469–73.

Alcabes P; Shoenbaum EE; Klein RS. Correlates of the rate of decline of CD4+ lymphocytes among injection drug users infected with the human immunodeficiency virus. *American Journal of Epidemiology* 1993, 137(9):989–1000.

Allen J; Murray A. Development of a neural network screening aid for diagnosing lower limb peripheral vascular disease from photoelectric plethysmography pulse waveforms. *Physiological Measurement*, 1993 Feb, 14(1):13–22.

Anderer P; Saletu B; Kloppel B; Semlitsch HV; Werner H. Discrimination between demented patients and normals based on topographic EEG slow wave activity: comparison between z statistics, discriminant analysis and artificial neural network classifiers. *Electroencephalography and Clinical Neurophysiology*, 1994 Aug, 91(2):108–17.

Anderson KM; Odell PM; Wilson PW; Kannel WB. Cardiovascular disease risk profiles. *American Heart Journal*, 1991 Jan, 121(1 Pt 2):293–8.

Aragon J; Weston; Warner R. Analysis of disease progression from clinical observations of US Air Force active duty members infected with the human immunodeficiency virus: Distribution of AIDS survival time from interval-censored observations. *Vaccine*, 1993, 11(5):552–4.

Arkes HR; Dawson NV; Speroff T; Harrell FE Jr; Alzola C; Phillips R; Desbiens N; Oye RK; Knaus W; Connors AF Jr. The covariance decomposition of the probability score and its use in evaluating prognostic estimates. SUPPORT Investigators. *Medical Decision Making*, 1995 Apr-Jun, 15(2):120–31.

Ashutosh K; Lee H; Mohan CK; Ranka S; Mehrotra K; Alexander C. Prediction criteria for successful weaning from respiratory support: statistical and connectionist analyses. *Critical Care Medicine*, 1992 Sep, 20(9):1295–301.

Astion ML; Wener MH; Thomas RG; Hunder GG; Bloch DA. Application of neural networks to the classification of giant cell arteritis. *Arthritis and Rheumatism*, 1994 May, 37(5):760–70.

Ballard D. Modular learning in hierarchical neural networks. In Schwartz EL (ed), *Computational Neuroscience*. Bradford, London, 1990.

Baxt WG. Analysis of the clinical variables driving decision in an artificial neural network trained to identify the presence of myocardial infarction. *Annals of Emergency Medicine*, 1992 Dec, 21(12):1439–44.

Baxt WG. Use of an artificial neural network for the diagnosis of myocardial infarction. *Annals of Internal Medicine*, 1991 Dec 1, 115(11):843–8.

Benediktsson JA; Sveinsson JR; Ersoy OK; Swain PH. Parallel consensual neural networks. In IEEE (ed), *Proceedings of 1993 IEEE International Conference on Neural Networks*, IEEE Service Center, Piscataway, 1993.

Bernstein IH. *Applied Multivariate Analysis*. Springer-Verlag, New York, 1988.

Bischof H. *Pyramidal Neural Networks*. Lawrence Erlbaum Associates, Mahwah, 1995.

Boddy L; Morris CW; Wilkins MF; Tarran GA; Burkill PH. Neural network analysis of flow cytometric data for 40 marine phytoplankton species. *Cytometry*, 1994 Apr 1, 15(4):283–93.

Bolinger RE; Hopfensperger KJ; Preston DF. Application of a virtual neurode in a model thyroid diagnostic network. In Clayton P(ed), *Proceedings of the Fifteenth Annual Symposium on Computer Applications in Medical Care*, McGraw-Hill, New York, 1991.

Bortolan G; Willems JL. Diagnostic ECG classification based on neural networks. *Journal of Electrocardiology*, 1993, 26 Suppl:75–9.

Bowerman BL; O'Connel RT. *Time Series Forecasting*. Duxbury Press, Boston, 1987.

Breiman L. *Classification and Regression Trees.* Wadsworth International Group, Belmont, 1984.

Breslow N. A generalized Kruskal-Wallis test for comparing K samples subject to unequal patterns of censorship. *Biometrika,* 1970, 57:579–94.

Buchman TG; Kubos KL; Seidler AJ; Siegforth MJ. A comparison of statistical and connectionist models for the prediction of chronicity in a surgical intensive care unit. *Critical Care Medicine*, 1994 May, 22(5):750–62.

Burke HB. Artificial neural networks for cancer research: outcome prediction. *Seminars in Surgical Oncology*, 1994 Jan-Feb, 10(1):73–9.

Cantoni V; Ferretti M. *Pyramidal Architectures for Computer Vision*. Plenum Press, New York, 1994.

Capurso A. Lipid metabolism and cardiovascular risk: should hypercholesterolemia be treated in the elderly? *Journal of Hypertension*. 1992 Apr, 10(2):S65–8.

Carpenter GA; Grossberg S. The ART of adaptive pattern recognition by a self-organizing neural network. *Computer*, 1988 Mar, 21(3):77–88.

Castelli WP; Anderson K; Wilson PW; Levy D. Lipids and risk of coronary heart disease. The Framingham Study. *Annals of Epidemiology*, 1992 Jan-Mar, 2(1–2):23–8.

Centers for Disease Control and Prevention. 1993 revised classification system for HIV infection and expanded surveillance case definition for AIDS among adolescents and adults. *MMWR,* 1992 Dec 18, 41(RR-17):1–19.

Centor RM. Signal detectability: The use of ROC curves and their analyses. *Medical Decision Making*, 1991, 11:102–6.

Chale JJ; Quantin C; Mosseri V; Asselain B; Moreau T; Dussere L. Testing the proportional hazards hypothesis on a tonsillar carcinoma data set: A Comparison of Methods. In Degoulet P; Piemme TE; Rienhoff O(eds)*, Proceedings of the Seventh World Congress on Medical Informatics*, North-Holland, Amsterdam, 1992.

Chang HH; Morse DL; Noonan C; Coles B; Mikl J; Rosen A; Putnam D; Smith PF. Survival and mortality patterns of an Acquired Deficiency Syndrome (AIDS) cohort in New York State. *American Journal of Epidemiology,* 1993, 138(5):341–9.

Cheng B; Titterington DM. Neural networks: A review from a statistical perspective. *Statistical Science*, 1994, 9(1):2–54.

Chiou YS; Lure YM. Hybrid lung nodule detection (HLND) system. *Cancer Letters*, 1994 Mar 15, 77(2-3):119–26.

Chlebowski RT; Grosvenor MB; Bernhard NH; Morales LS; Bulcavage LM. Nutritional status, gastrointestinal dysfunction, and survival in patients with

AIDS. *American Journal of Gastroenterology*, 1989 Oct, 84(10):1288–93.

Cho S; Kim JH. Hierarchically structured neural networks for printed Hangul character recognition. In IEEE (ed), *International Joint Conference on Neural Networks*, IEEE, New York, 1990.

Christensen R. *Log-Linear Models*. Springer-Verlag, New York, 1990.

Collet D. *Modelling Survival Data in Medical Research*. Chapman and Hall, London, 1994.

Cormen TH; Leiserson CE; Rivest RL. *Introduction to Algorithms*, MIT Press, Cambridge, 1990.

Cox DR. *Analysis of Survival Data*. Chapman and Hall, London, 1984.

Cox DR. Regression models and life-tables. *Journal of the Royal Statistics Society B*, 1972, 34:187–202.

Cupples LA; D'Agostino RB; Anderson K; Kannel WB. Comparison of baseline and repeated measure covariate techniques in the Framingham Heart Study. *Statistics in Medicine*, 1988 Jan-Feb, 7(1-2):205–22.

Curry B; Rumelhart DE. MSnet: A neural network that classifies mass spectra. *Tetrahedron Computer Methodology*, 1990, 3:213–37.

Curtis JR; Patrick DL. Race and survival time with AIDS: A synthesis of the literature. *American Journal of Public Health*, 1993 Oct, 83(10):1425–8.

Cutler SJ; Ederer E. Maximum utilization of the life-table in analysis of survival. *Journal of Chronic Diseases*, 1958, 6:699–712.

D'Agostino RB; Lee ML; Belanger AJ; Cupples LA; Anderson K; Kannel WB. Relation of pooled logistic regression to time dependent Cox regression analysis: The Framingham Heart Study. *Statistics in Medicine*, 1990 Dec, 9(12):1501–15.

Dannemann B; McCutchan JA; Israelski D; Antoniskis D; Leport C; Luft B; Nussbaum J; Clumeck N; Morlat P; Chiu J. Treatment of toxoplasmic encephalitis in patients with AIDS. A randomized trial comparing pyrimethamine plus clindamycin to pyrimethamine plus sulfadiazine. The California Collaborative Treatment Group. *Annals of Internal Medicine*, 1992 Jan 1, 116(1):33–43.

Dawber TR. *The Framingham Study: The Epidemiology of Atherosclerotic Disease*. Harvard University Press, Cambridge, 1980.

Devine B; Macfarlane PW. Detection of electrocardiographic 'left ventricular strain' using neural nets. *Medical and Biological Engineering and Computing*, 1993 Jul, 31(4):343–8.

Doig GS; Inman KJ; Sibbald WJ; Martin CM; Robertson JM. Modeling mortality in the intensive care unit: comparing the performance of a back-propagation, associative-learning neural network with multivariate logistic regression. in In Safran C(ed), *Proceedings of the Seventeenth Annual Symposium on Computer Applications in Medical Care*, McGaw-Hill, New York, 1993.

Doyle HR; Dvorchik I; Mitchell S; Marino IR; Ebert FH; McMichael J; Fung JJ. Predicting outcomes after liver transplantation. A connectionist approach. *Annals of Surgery,* 1994 Apr, 219(4):408–15.

Drown DJ; Engler MM. New guidelines for blood cholesterol by the National Cholesterol Education Program (NCEP). National Cholesterol Education Program (NCEP). *Progress in Cardiovascular Nursing*, 1994 Winter, 9(1):43–4.

Duda RO; Hart PE. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

Easterbrook PJ; Chmiel JS; Hoover DR; Saah AJ; Kaslow RA; Kingsley LA;

Detels R. Racial and ethnic differences in human immunodeficiency virus type 1 (HIV-1) seroprevalence among homosexual and bisexual men. The Multicenter AIDS Cohort Study. *American Journal of Epidemiology*, 1993 Sep 15, 138(6):415–29.

Easterbrook PJ; Keruly JC; Creagh-Kirk T; Richman DD; Chaisson RE; Moore RD. Racial and ethnic differences in outcome in zidovudine-treated patients with advanced HIV disease. Zidovudine Epidemiology Study Group. *JAMA*, 1991 Nov 20, 266(19):2713–8.

Ebell MH. Artificial neural networks for predicting failure to survive following in-hospital cardiopulmonary resuscitation. *Journal of Family Practice*, 1993 Mar, 36(3):297–303.

Edenbrandt L; Heden B; Pahlm O. Neural networks for analysis of ECG complexes. *Journal of Electrocardiology*, 1993, 26 Suppl:74.

Efron B. Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of the American Statistical Association*, 1983, 78(382):316–31.

Efron B. Bootstrap methods: Another look at the jackknife. *Annals of Statistics*, 1979, 7:1–26.

Egan JP. *Signal Detection Theory and ROC Analysis*. Academic Press, New York, 1975.

Evans SJ; Hastings H; Bodenheimer MM. Differentiation of beats of ventricular and sinus origin using a self-training neural network. *Pacing and Clinical Electrophysiology*, 1994 Apr, 17(4 Pt 1):611–26.

Fagan LM. *VM: Representing Time-Dependent Relations in a Clinical Setting* (Ph.D. Dissertation). Department of Computer Science, Stanford University, Stanford, 1980.

Fahey JL; Taylor JM; Detels R; Hofmann B; Melmed R; Nishanian P; Giorgi JV. The prognostic value of cellular and serologic markers in infection with human immunodeficiency virus type 1. *New England Journal of Medicine*, 1990 Jan 18, 322(3):166–72.

Fahlman SE. Faster-learning variations on backpropagation: An empirical study. In Touretzky D; Hinton G; Sejnowski T (eds), *Proceedings of the 1988 Connectionist Summer School*, Morgan-Kaufmann, San Mateo, 1988.

Fischl MA; Richman DD; Grieco MH; Gottlieb MS; Volberding PA; Laskin OL; Leedom JM; Groopman JE; Mildvan D; Shooley RT; Jackson GG; Durack DT; King D; The AZT Collaborative Working Group. The efficacy of azidothymidine (AZT) in the treatment of patients with AIDS and AIDS-related complex: A double-blind, placebo-controlled trial. *New England Journal of Medicine*, 1987 July 23, 317(4):186–91.

Freeman DH. *Applied Categorical Data Analysis*. Dekker, New York, 1987.

Freeman H. Machine vision approaches to automatic inspection. *Progress in Image Analysis and Processing,* vol 2, Academic Press, San Diego, 1988.

Freeman R; Goodacre R; Sisson PR; Magee JG; Ward AC; Lightfoot NF. Rapid identification of species within the *Mycobacterium tuberculosis* complex by artificial neural network analysis of pyrolysis mass spectra. *Journal of Medical Microbiology*, 1994 Mar, 40(3):170–3.

Freund KM; Belanger AJ; D'Agostino RB; Kannel WB. The health risks of smoking. The Framingham Study: 34 years of follow-up. *Annals of Epidemiology*, 1993 Jul, 3(4):417–24.

Fries J. *An HIV Data Resource for Systematic Health Policy Study* (grant proposal). Agency for Health Care Policy and Research, Rockville, 1992.

Fuchs D; Spira TJ; Hausen A; Reibnegger G; Werner ER; Felmayer GW; Wachter H. Neopterin as a predictive marker for disease progression in human immunodeficiency virus type 1 infection. *Clinical Chemistry*, 1989 Aug, 35(8):1746–9.

Fukushima K. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Networks*, 1988, 1:119–30.

Gallant, S.I. Connectionist expert systems. *Communications of the ACM*, Feb. 1988, 31(2):152–69.

Gagnon DR; Zhang TJ; Brand FN; Kannel WB. Hematocrit and the risk of cardiovascular disease—the Framingham study: a 34-year follow-up. *American Heart Journal*, 1994 Mar, 127(3):674–82.

Garber AM. Personal communication, 1994.

Gardner LI Jr; Brundage JF; McNeil JG; Milazzo MJ; Redfield RR; Aronson NE; Craig DB; Davis C; Gates RH; Levin LI. Predictors of HIV-1 disease progression in early- and late-stage patients: The U.S. Army Natural History Cohort. Military Medical Consortium for Applied Retrovirology. *Journal of Acquired Immune Deficiency Syndromes*, 1992, 5(8):782–93.

Gillman MW; Kannel WB; Belanger A; D'Agostino RB. Influence of heart rate on mortality among persons with hypertension: the Framingham Study. *American Heart Journal*, 1993 Apr, 125(4):1148–54.

Glantz SA. *Primer of Applied Regression and Analysis of Variance*. McGraw-Hill, New York, 1990.

Goodman P. Personal communication, 1994.

Graham NM; Zeger SL; Park LP; Phair JP; Detels R; Vermund SH; Ho M; Saah AJ. Effect of zidovudine and *Pneumocystis carinii* pneumonia prophylaxis on progression of HIV-1 infection to AIDS. The Multicenter AIDS Cohort Study. *Lancet*, 1991 Aug 3, 338(8762):265–9.

Gray NAB. Constraints on learning machine classification methods. *Analytical Chemistry*, 1976, 48(14):2265–8.

Griffin MP; Scallion DF; Moorman JR. The dynamic range of neonatal heart rate variability. *Journal of Cardiovascular Electrophysiology*, 1994 Feb, 5(2):112–24.

Hanley JA; McNeil BJ. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology*, 1983 Sep, 148(3):839–43.

Hanley JA; McNeil BJ. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 1982 Apr, 143(1):29–36.

Hanson DL; Horsburgh CR Jr; Fann SA; Havlik JA; Thompson SE 3d. Survival prognosis of HIV-infected patients. *Journal of Acquired Immune Deficiency Syndromes*, 1993 Jun, 6(6):624–9.

Harris JE. Improved short-term survival of AIDS patients initially diagnosed with *Pneumocystis carinii* pneumonia, 1984 through 1987. *JAMA*, 1990 Jan 19, 263(3):397–401.

Hebb D. *The Organization of Behavior.* Wiley, New York, 1949.

Heden B; Edenbrandt L; Haisty WK Jr; Pahlm O. Artificial neural networks for the electrocardiographic diagnosis of healed myocardial infarction. *American Journal of Cardiology*, 1994 Jul 1, 74(1):5–8.

Hertz JA; Palmer RG; Krogh, AS. *Introduction to the Theory of Neural Computation*. Addison-Wesley, Redwood City, 1991.

Higgins M; D'Agostino R; Kannel W; Cobb J. Benefits and adverse effects of weight loss. Observations from the Framingham Study. *Annals of Internal Medicine*, 1993 Oct 1, 119(7 Pt 2):758–63.

Hilden J. The area under the ROC curve and its competitors. *Medical Decision Making*, 1991, 11:95–101.

Ho KK; Pinsky JL; Kannel WB; Levy D. The epidemiology of heart failure: the Framingham Study. *Journal of the American College of Cardiology*, 1993 Oct, 22(4 Suppl A):6A–13A.

Ho KK; Anderson KM; Kannel WB; Grossman W; Levy D. Survival after the onset of congestive heart failure in Framingham Heart Study subjects. *Circulation*, 1993 Jul, 88(1):107–15.

Hopfield JJ. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences of the United States of America*, 1984 May, 81(10):3088–92.

Hornik K; Stichcombe M; White H. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–66, 1989.

Hripcsak G. Using connectionist modules for decision support. *Methods of Information in Medicine*, 1990, 29:167–81.

Hrycej T. *Modular Learning in Neural Networks*. John Wiley and Sons, New York, 1992.

Hudgins B; Parker P; Scott RN. A new strategy for multifunction myoelectric control. *IEEE Transactions on Biomedical Engineering*, 1993 Jan, 40(1):82–94.

Ingram DD; Kleinman JC. Empirical comparisons of proportional hazards and logistic regression models. *Statistics in Medicine*, 1989 May, 8(5):525–38.

Jando G; Siegel RM; Horvath Z; Buzsaki G. Pattern recognition of the electroencephalogram by artificial neural networks. *Electroencephalography and Clinical Neurophysiology*, 1993 Feb, 86(2):100–9.

Jenner JL; Ordovas JM; Lamon-Fava S; Schaefer MM; Wilson PW; Castelli WP; Schaefer EJ. Effects of age, sex, and menopausal status on plasma lipoprotein(a) levels. The Framingham Offspring Study. *Circulation*, 1993 Apr, 87(4):1135–41.

Jordan RA; Nowlan SJ; Hinton SJ. Adaptive mixtures of local experts. *Neural Computation*, 1991, 3:79–87.

Justice AC; Feinstein AR; Wells CK. A new prognostic staging system for the acquired immunodeficiency syndrome. *New England Journal of Medicine*, 1989 May 25, 320(21):1388–93.

Kammerer BR; Kupper WA. Experiments for isolated-word recognition with single- and two-layer perceptrons. *Neural Networks*, 1990, 3(6):693–706.

Kannel WB; Larson M. Long-term epidemiologic prediction of coronary disease. The Framingham experience. *Cardiology*, 1993, 82(2–3):137–52.

Kannel WB. Hypertension as a risk factor for cardiac events—epidemiologic results of long-term studies. *Journal of Cardiovascular Pharmacology*, 1993, 21 Suppl 2:S27–37.

Kannel WB; D'Agostino RB; Belanger AJ. Update on fibrinogen as a cardiovascular risk factor. *Annals of Epidemiology*, 1992 Jul, 2(4):457–66.

Kannel WB; Cupples LA; Ramaswami R; Stokes J 3d; Kreger BE; Higgins M. Regional obesity and risk of cardiovascular disease; the Framingham Study. *Journal of Clinical Epidemiology*, 1991, 44(2):183–90.

Kannel WB; Gagnon DR; Cupples LA. Epidemiology of sudden coronary death:

population at risk. *Canadian Journal of Cardiology*, 1990 Dec, 6(10):439–44.

Kannel WB; D'Agostino RB; Wilson PW; Belanger AJ; Gagnon DR. Diabetes, fibrinogen, and risk of cardiovascular disease: the Framingham experience. *American Heart Journal*, 1990 Sep, 120(3):672–6.

Kannel WB; Higgins M. Smoking and hypertension as predictors of cardiovascular risk in population studies. *Journal of Hypertension.* 1990 Sep, 8(5):S3–8.

Kaplan EL; Meier P. Nonparametric estimation from incomplete observations. *American Statistical Association Journal*, 1958 June, 457–81.

Kappen HJ; Neijt JP. Advanced ovarian cancer. Neural network analysis to predict treatment outcome. *Annals of Oncology*, 1993, 4 Suppl 4:S31–4.

Katz S; Katz AS; Lowe N; Quijano RC. Neural net-bootstrap hybrid methods for prediction of complications in patients implanted with artificial heart valves. *Journal of Heart Valve Disease*, 1994 Jan, 3(1):49–52.

Katz AS; Katz S; Wickham E; Quijano RC. Prediction of valve-related complications for artificial heart valves using adaptive neural networks: a preliminary study. *Journal of Heart Valve Disease*, 1993 Sep, 2(5):504–8.

Kazmierczak SC; Catrou PG; Van Lente F. Diagnostic accuracy of pancreatic enzymes evaluated by use of multivariate data analysis. *Clinical Chemistry*, 1993 Sep, 39(9):1960–5.

Kleinbaum DG. *Applied Regression Analysis and Other Multivariable Method*s. PWS-Kent, Boston, 1988.

Kloppel B. Application of neural networks for EEG analysis. Considerations and first results. *Neuropsychobiology*, 1994, 29(1):39–46.

Knaus WA; Wagner DP; Draper EA; Zimmerman JE; Bergner M; Bastos PG; Sirio CA; Murphy DJ; Lotring T; Damiano A. The APACHE III prognostic system. Risk prediction of hospital mortality for critically ill hospitalized adults. *Chest*, 1991 Dec, 100(6):1619–36.

Kohonen T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 1982, 43:59–69.

Kreger BE; Odell PM; D'Agostino RB; Wilson PF. Long-term intraindividual cholesterol variability: natural course and adverse impact on morbidity and mortality—the Framingham Study. *American Heart Journal*, 1994 Jun, 127(6):1607–14.

Kronmal RA; Cain KC; Ye Z; Omenn GS. Total serum cholesterol levels and mortality risk as a function of age. A report based on the Framingham data. *Archives of Internal Medicine*, 1993 May 10, 153(9):1065–73.

Larsen RJ; Morris LM. *Statistics*. Prentice Hall, Englewood Cliffs, 1990.

Laursen P. Event detection on patient monitoring data using Causal Probabilistic Networks. *Methods of Information in Medicine*, 1994 Mar, 33(1):111–5.

Lee ET. *Statistical Methods for Survival Data Analysis*. John Wiley and Sons, New York, 1992.

Lemp GF; Payne SF; Neal D; Temelso T; Rutherford GW. Survival trends for patients with AIDS. *JAMA*, 1990 Jan 19, 263(3):402–6.

Leon MA; Rasanen J; Mangar D. Neural network-based detection of esophageal intubation. *Anesthesia and Analgesia*, 1994 Mar, 78(3):548–53.

Levine AM; Sullivan-Halley J; Pike MC; Rarick MU; Loureiro C; Bernstein-Singer M; Willson E; Brynes R; Parker J; Rasheed S; Gill PS. Human immunodeficiency virus-related lymphoma: Prognostic factors predictive of survival. *Cancer*, 1991 Dec 1, 6:2466–72.

Lew RA; Day CL; Harrist TJ; Wood WC; Mihm Jr MC. Multivariate analysis: Some guidelines for physicians. *JAMA*, Feb 4, 1983, 249(5):641–3.

Libman H.Pathogenesis, natural history, and classification of HIV infection. *Primary Care; Clinics in Office Practice*, 1992 Mar, 19(1):1–17.

Lin JS; Ligomenides PA; Freedman MT; Mun SK. Application of artificial neural networks for reduction of false-positive detections in digital chest radiographs. In Safran C(ed), *Proceedings of the Seventeenth Annual Symposium on Computer Applications in Medical Care*, McGraw-Hill, New York, 1993.

Lin RY; Goodhart P. The role of oral candidiasis in survival and hospitalization patterns: Analysis of an inner city hospital human immunodeficiency virus/acquired immune deficiency syndrome registry. *American Journal of the Medical Sciences*, 1993 Jun, 305(6):345–53.

Lo SC; Freedman MT; Lin JS; Mun SK. Automatic lung nodule detection using profile matching and back-propagation neural network techniques. *Journal of Digital Imaging*, 1993 Feb, 6(1):48–54.

Longini IM Jr; Clark WS; Gardner LI; Brundage JF. The dynamics of CD4+ T-lymphocyte decline in HIV-infected individuals: A Markov modeling approach. *Journal of Acquired Immune Deficiency Syndromes*, 1991, 4(11):1141–7.

Longini IM Jr; Clark WS; Byers RH; Ward JW; Darrow WW; Lemp GF; Hethcote HW. Statistical analysis of the stages of HIV infection using a Markov model. *Statistics in Medicine*, 1989 Jul, 8(7):831–43.

Lowe D; Webb AR. Exploiting prior knowledge in network optimization: An illustration from medical prognosis. *Network*, 1990, 1:299–323.

Lubeck DP; Bennett CL; Mazonson PD; Fifer SK; Fries JF. Quality of life and health service use among HIV-infected patients with chronic diarrhea. *Journal of Acquired Immune Deficiency Syndromes*, 1993 May, 6(5):478–84.

Lubeck DP; Fries JF. Health status among persons infected with human immunodeficiency virus. A community-based study. *Medical Care*, 1993 Mar, 31(3):269–76.

Lubeck DP; Fries JF. Changes in quality of life among persons with HIV infection. *Quality of Life Research*, 1992 Dec, 1(6):359–66.

Maclin PS; Dempsey J. How to improve a neural network for early detection of hepatic cancer. *Cancer Letters*, 1994 Mar 15, 77(2-3):95–101.

Maclin PS; Dempsey J. Using an artificial neural network to diagnose hepatic masses. *Journal of Medical Systems*, 1992 Oct, 16(5):215–25.

Mantel N; Haenszel W. Statistical aspects of the analysis of data from retrospective studies of disease. *Journal of the National Cancer Institute*, 1959, 22, 719–48.

Mariotto AB; Mariotti S; Pezzotti P; Rezza G; Verdecchia A. Estimation of the acquired immunodeficiency syndrome incubation period in intravenous drug users: A comparison with male homosexuals. *American Journal of Epidemiology*, 1992 Feb 15, 135(4):428–37.

Maron MJ. *Numerical Analysis: A Practical Approach*. MacMillan, New York, 1987.

Matsuoka T; Hamada H; Nakatsu R. Syllable recognition using integrated neural networks. In IEEE (ed), *International Joint Conference on Neural Networks*, IEEE TAB Neural Network Committee, New York, 1989.

McAuliffe JD. Data compression of the exercise ECG using a Kohonen neural network. *Journal of Electrocardiology*, 1993, 26 Suppl:80–9.

McCullough JM; Pitts W. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 1943, 5:115–133.

McGonigal MD; Cole J; Schwab CW; Kauder DR; Rotondo MF; Angood PB. A new approach to probability of survival scoring for trauma quality assurance. *Journal of Trauma*, 1993 Jun, 34(6):863–8.

McGuire WL; Tandon AK; Allred DC; Chamness GC; Ravdin PM; Clark GM. Treatment decisions in axillary node-negative breast cancer patients. *Monographs / National Cancer Institute*, 1992(11):173–80.

McNeil BJ; Hanley JA. Statistical approaches to the analysis of receiver operating characteristic (ROC) curves. *Medical Decision Making*, 1984, 11:137–50.

McShane DJ; Fullerton S; Salehi T; Mathews JK; Bloch D; Mathews WC; Lipil WH; Covington DD. AIDS survival: Private and public-insured patients in ATHOS. In *Proceedings of the IX International Conference on AIDS*. Berlin, 1993.

Mead C. Silicon models of neural computation. *IEEE First International Conference on Neural Networks*, vol 1:93–106, 1987.

Medsker L. *Hybrid Neural Networks and Expert Systems*. Kluwer Academic, Boston, 1994

Michalski RS; Chilausky RL. Knowledge acquisition by encoding expert rules versus computer induction from examples: A case study involving soybean pathology. *International Journal of Man-Machine Studies*, 1980, 12:63–87.

Minsky M; Papert S. *Perceptrons; an Introduction to Computational Geometry.* MIT Press, Cambridge, 1969.

Moise A; Clement B; Raissis M; Nanopoulos P. A test for crossing receiver operating characteristic (ROC) curves. *Commun Statist Theory Meth*, 1988, 17:1985–2003.

Moore RD; Hidalgo J; Sugland BW; Chaisson RE. Zidovudine and the natural history of the acquired immunodeficiency syndrome. *New England Journal of Medicine*, 1991 May 16, 324(20):1412–6.

Moore RD; Keruly J; Richman DD; Creagh-Kirk T; Chaisson RE; The Zidovudine Epidemiology Study Group. Natural history of advanced HIV disease in patients treated with zidovudine. *AIDS* 1992, 6(97):671–7.

Murphy PM; Aha DW. *UCI Repository of Machine Learning Databases* (on-line directory). University of California at Irvine, Department of Information and Computer Science, Irvine, 1993.

Musen MA. Automated support for building and extending expert models. *Machine Learning*, Dec. 1989, 4(3-4):347–75.

Ohno-Machado L; Musen MA. Hierarchical neural networks for partial diagnosis in medicine. In INNS(ed), *Proceedings of the 1994 World Congress on Neural Networks*. Lawrence Erlbaum Associates, Hillsdale, 1994.

Ohno-Machado L. Identification of low frequency patterns in backpropagation neural networks. In Ozbolt JG(ed), *Proceedings of the Eighteenth Symposium on Computer Applications in Medical Care*, Hanley and Belfus, Philadelphia, 1994.

Pacala JT; McBride PE; Gray SL. Management of older adults with hypercholesterolaemia. *Drugs and Aging*, 1994 May, 4(5):366–78.

Pedersen C; Kolby P; Sindrup J; Gaub J; Ullman S; Gerstoft J; Lindhardt BO; Dickmeiss E. The development of AIDS or AIDS-related conditions in a cohort of HIV antibody-positive homosexual men during a 3-year follow-up period. *Journal of Internal Medicine*, 1989 Jun, 225(6):403–9.

Percy DF; Hine TJ. Multivariate analysis of cholesterol distribution for monitoring the risk of coronary heart disease. *Statistics in Medicine*, 1993 May 30, 12(10):967–74.

Peretto P. *An Introduction to the Modeling of Neural Networks*. Cambridge University Press, Cambridge, 1992.

Piette JD; Intrator O; Zierler S; Mor V; Stein MD. An exploratory analysis of survival with AIDS using a nonparametric tree-structured approach. *Epidemiology*, 1992 Jul, 3(4):310–8.

Posner BM; Cupples LA; Miller DR; Cobb JL; Lutz KJ; D'Agostino RB. Diet, menopause, and serum cholesterol levels in women: the Framingham Study. *American Heart Journal*, 1993 Feb, 125(2 Pt 1):483–9.

Presnell SR; Cohen BI; Cohen FE. MacMatch: a tool for pattern-based protein secondary structure prediction. *Computer Applications in the Biosciences*, 1993 Jun, 9(3):373–4.

Pritchard WS; Duke DW; Coburn KL; Moore NC; Tucker KA; Jann MW; Hostetler RM. EEG-based, neural-net predictive classification of Alzheimer's disease versus control subjects is augmented by non-linear EEG measures. *Electroencephalography and Clinical Neurophysiology*, 1994 Aug, 91(2):118–30

Quinlan JR. Induction of decision trees. *Machine Learning*, 1986, 1:8–106.

Quinn TC; Narain JP; Zacarias FR. AIDS in the Americas: A public health priority for the region. *AIDS*, 1990 Aug, 4(8):709–24.

Rabeneck L; Crane MM; Risser JM; Lacke CE; Wray NP. A simple clinical staging system that predicts progression to AIDS using CD4 count, oral thrush, and night sweats. *Journal of General Internal Medicine*, 1993 Jan, 8(1):5–9.

Ravdin PM; Clark GM; Hough JJ; Owens MA; McGuire WL. Neural network analysis of DNA flow cytometry histograms. *Cytometry*, 1993, 14(1):74–80.

Ravdin PM; Clark GM. A practical application of neural network analysis for predicting outcome of individual breast cancer patients. *Breast Cancer Research and Treatment*, 1992, 22(3):285–93.

Ravdin PM; Clark GM; Hilsenbeck SG; Owens MA; Vendely P; Pandian MR; McGuire WL. A demonstration that breast cancer recurrence can be predicted by neural network analysis. *Breast Cancer Research and Treatment*, 1992, 21(1):47–53.

Redelmeier DA; Bloch DA; Hickam DH. Assessing predictive accuracy: how to compare Brier scores. *Journal of Clinical Epidemiology*, 1991, 44(11):1141–6.

Reggia JA. Neural computation in medicine. *Artificial Intelligence in Medicine*, 1993 Apr, 5(2):143–57.

Reibnegger G; Spira TJ; Fuchs D; Werner-Felmayer G; Dierich MP; Wachter H. Individual probability for onset of full-blown disease in patients infected with human immunodeficiency virus type 1. *Clinical Chemistry*, 1991 Mar, 37(3):351–5.

Rinast E; Linder R; Weiss HD. Neural network approach for computer-assisted interpretation of ultrasound images of the gallbladder. *European Journal of Radiology*, 1993 Nov, 17(3):175–8.

Rogers SK; Ruck DW; Kabrisky M. Artificial neural networks for early detection and diagnosis of cancer. *Cancer Letters*, 1994 Mar 15, 77(2-3):79–83.

Rosenblatt F. *Principals of Neurodynamics*. Spartan, New York, 1962.

Rothenberg, R; Woefel, M; Stoneburner, R; Milberg, J; Parker, R; Truman, B.

Survival with the acquired immunodeficiency syndrome: Experience with 5833 cases in New York City. *New England Journal of Medicine*, 1987 Nov 19, 317(21):1297–1302.

Rubsamen-Waigmann H; Schroder B; Biesert L; Bauermeister CD; von Briesen H; Suhartono H; Zimmermann F; Brede HD; Regeniter A; Gerte S. Markers for HIV-disease progression in untreated patients and patients receiving AZT: Evaluation of viral activity, AZT resistance, serum cholesterol, beta 2-microglobulin, CD4+ cell counts, and HIV antigen. *Infection*, 1991, 19 Suppl 2:S77–82.

Rumelhart DE. Backpropagation: theory, architectures, and applications. Lawrence Erlbaum Associates, Hillsdale, 1995.

Rumelhart DE; Hinton GE; Williams RJ. Learning internal representation by error propagation. In Rumelhart, D.E., and McClelland, J.L. (eds) *Parallel Distributed Processing*. MIT Press, Cambridge, 1986.

Saah AJ; Munoz A; Kuo V; Fox R; Kaslow RA; Phair JP; Rinaldo CR Jr; Detels R; Polk BF. Predictors of the risk of development of acquired immunodeficiency syndrome within 24 months among gay men seropositive for human immunodeficiency virus type 1: A report from the Multicenter AIDS Cohort Study. *American Journal of Epidemiology*, 1992 May 15, 135(10):1147–55.

SAS Institute. *SAS/STAT User's Guide*, Version 6, Fourth Edition. SAS Institute Inc., Cary, 1990.

Schechter MT; Craib KJ; Le TN; Montaner JS; Douglas B; Sestak P; Willoughby B; O'Shaughnessy MV. Susceptibility to AIDS progression appears early in HIV infection. *AIDS*, 1990 Mar, 4(3):185–90.

Seage GR; Oddleifson S; Carr E; Shea B; Makarewicz-Robert L; van Beuzekom M; De Maria A. Survival with AIDS in Massachusetts, 1979 to 1989. *American Journal of Public Health*, 1993, 83(1):72–8.

Seage GR; Landers S; Lamb GA; Epstein AM. Effect of changing patterns of care and duration of survival on the cost of treating the acquired immunodeficiency syndrome (AIDS). *American Journal of Public Health*, 1990 Jul, 80(7):835–9.

Segal MR; Bloch DA. A comparison of estimated proportional hazards models and regression trees. *Statistics in Medicine*, 1989, 8:539–50.

Selvin S. *Statistical Analysis of Epidemiological Data*. Oxford University Press, New York, 1991.

Selwyn PA; Alcabes P; Hartel D; Buono D; Schoenbaum EE; Klein RS; Davenny K; Friedland GH. Clinical manifestations and predictors of disease progression in drug users with human immunodeficiency virus infection. *New England Journal of Medicine*, 1992 Dec 10, 327(24):1697–703.

Sharpe PK; Solberg HE; Rootwelt K; Yearworth M. Artificial neural networks in diagnosis of thyroid function from in vitro laboratory tests. *Clinical Chemistry*, 1993 Nov, 39(11 Pt 1):2248–53.

Shortliffe EH; Perreault LE; Fagan LM; Wiederhold G. *Medical Informatics: Computer Applications in Medicine*. Addison-Wesley, Reading, 1990.

Shortliffe EH. *Computer-Based Medical Consultation: MYCIN*. American Elsevier, New York, 1976.

Somoza E; Somoza JR. A neural-network approach to predicting admission decisions in a psychiatric emergency room. *Medical Decision Making*, 1993 Oct-Dec, 13(4):273–80.

Stone M. Cross-validation: A review. *Math Operationforsch Statist*, 1977, 9:127–39.

Swets JA. *Evaluation of Diagnostic Systems*. Academic Press, London, 1982.

Swets JA. The relative operating characteristic in psychology. *Science*, 1973, 182:990.

Tarone RE; Ware J. On distribution-free tests for equality of survival distributions. *Biometrika*, 1979, 64:156–60.

Temple NJ; Walker AR. Blood cholesterol and coronary heart disease: changing perspectives. *Journal of the Royal Society of Medicine*, 1994 Aug, 87(8):450–3.

Tu JV; Guerriere MR. Use of a neural network as a predictive instrument for length of stay in the intensive care unit following cardiac surgery. *Computers and Biomedical Research*, 1993 Jun, 26(3):220–9.

U.S. Preventive Task Force. *Guide to Clinical Preventive Services: An Assessment of the Effectiveness of 169 Interventions*. William and Wilkins, Baltimore, 1989.

Ullman JD. *Foundations of Computer Science*, Computer Science Press, New York, 1993.

Vadhan-Raj S; Wong; G; Gnecco; C; Cunningham-Rundles S; Krim M; Real FX; Oettgen HF; Krown SE. Immunological variables as predictors of prognosis in patients with Kaposi's sarcoma and the acquired immunodeficiency syndrome. *Cancer Research*, 1986 Jan, 46:417–25.

Veronesi R. *Doenças Infecciosas e Parasitárias*. Guanabara-Koogan, Rio de Janeiro, 1992.

Walker MG. *Probability Estimation for Classification Trees and DNA Sequence Analysis* (Ph.D. dissertation). Department of Computer Science, Stanford University, Stanford, 1992.

Weigend AS; Rumelhart DE; Huberman BA. Generalization by weight-elimination applied to currency exchange rate prediction. In IEEE(ed), *International Joint Conference on Neural Networks.* IEEE, New York, 1991.

Weinstein JN; Myers T; Buolamwini J; Raghavan K; van Osdol W; Licht J; Viswanadhan VN; Kohn KW; Rubinstein LV; Koutsoukos AD; et al. Predictive statistics and artificial intelligence in the U.S. National Cancer Institute's Drug Discovery Program for Cancer and AIDS. *Stem Cells*, 1994 Jan, 12(1):13–22.

Wenger, JD; Whalen, CC; Lederman, MM; Spech, TJ; Carey, JT; Tomford, JW; Landefeld, CS. Prognostic factors in acquired immunodeficiency syndrome. *Journal of General Internal Medicine*, 3:464–70, 1988.

Whittle H; Egboga A; Todd J; Corrah T; Wilkins A; Demba E; Morgan G; Rolfe M; Berry N; Tedder R. Clinical and laboratory predictors of survival in Gambian patients with symptomatic HIV-1 or HIV-2 infection. *AIDS*, 1992 Jul, 6(7):685–9.

Widrow B; Hoff ME. Adaptive switching circuits. In *1960 IRE WECON Convention Record*. IRE, New York, 1960.

Wilding P; Morgan MA; Grygotis AE; Shoffner MA; Rosato EF. Application of backpropagation neural networks to diagnosis of breast and ovarian cancer. *Cancer Letters*, 1994 Mar 15, 77(2-3):145–53.

Wong ND; Wilson PW; Kannel WB. Serum cholesterol as a prognostic factor after myocardial infarction: the Framingham Study. *Annals of Internal Medicine*, 1991 Nov 1, 115(9):687–93.

Wu FY; Slater JD; Honig LS; Ramsay RE. A neural network design for event-related potential diagnosis. *Computers in Biology and Medicine*, 1993 May, 23(3):251–64.

Yang TF; Devine B; Macfarlane PW. Deterministic logic versus software-based artificial neural networks in the diagnosis of atrial fibrillation. *Journal of Electrocardiology*, 1993, 26 Suppl:90–4.

Yates JF. External correspondence: decompositions of the mean probability score. *Org Behav Human Perf* 1982, 30, 132–56.

Zhang W; Doi K; Giger ML; Wu Y; Nishikawa RM; Schmidt RA. Computerized detection of clustered microcalcifications in digital mammograms using a shift-invariant artificial neural network. *Medical Physics*, 1994 Apr, 21(4):517–24.