Neural networks in economic modelling

Verkooijen, W.J.H.

Publication date:
1996

CentER
for
Economic Research

Neural Networks in Economic Modelling

An Empirical Study

William J.H. Verkooijen

Tilburg University

Neural Networks

in

Economic Modelling

Neural Networks
in
Economic Modelling
An Empirical Study

Proefschrift

ter verkrijging van de graad van doctor aan de Katholieke Universiteit Brabant, op gezag van de rector magnificus, prof. dr. L.F.W. de Klerk, in het openbaar te verdedigen ten overstaan van een door het college van dekanen aangewezen commissie in de aula van de Universiteit op

vrijdag 9 februari 1996 om 16.15 uur

door

Wilhelmus Johannes Henricus Verkooijen

geboren op 8 januari 1968 te Oosterhout (NB)

PROMOTOREN: Prof. Dr. Ir. H.A.M. Daniels
Prof. Dr. J.E.J. Plasmans

AAN SUSANNE EN MIJN OUDERS

Acknowledgements

At this place I want to thank all people who have, somehow or other, contributed to the realization of this dissertation. Some of them I would like to mention explicitly.

First, I want to thank my promotor Hennie Daniels for the intensive support in the last four years and for the close involvement in the research. I also want to thank my second promotor Joseph Plasmans for the stimulating discussions and for sharing his economic knowledge.

I want to thank the partners in the SPES-project for providing a scientific environment in which I could present my research; each meeting was a valuable and enjoyable experience.

Further, I want to thank Jack Kleijnen, Louis Pau, and Geoff Wyatt for carefully proofreading the manuscript. Their suggestions have certainly improved the manuscript.

I also want to thank my ex-roommates Ad Feelders, Chad Ekering, and Pieter Swinkels for their company during the last four years; additionally, I want to thank Ad for the many lengthy discussions and for keeping me company on various intercontinental flights to conferences on AI and Economics.

Last but not least, I want to thank Susanne for the invaluable support and patience during the last four years.

Contents

Introduction

A problem common to many disciplines, in particular economics, is that of approximating a variable of interest Y by a function of some variables X, given only the value Y of the function (often perturbed by noise) at various points in the X-space. For example, X could contain measurements of various economic indicators in a particular month and Y could be the monthly average of the dollar-deutschmark exchange rate. The variables of X are often referred to as predictor, input, explanatory, or independent variables, while the variables of Y often go by names such as response, output or dependent variables.

In the literature this problem goes under various different names, such as multiple (or multivariate) regression, curve fitting, and learning. Research on the regression problem occurs in several scientific areas: applied mathematics, statistics, econometrics, computer science, and engineering. The basic condition for regression is that repeated measurements on X and Y, $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, can be made, which allows us to build up a form of empirical knowledge about the phenomenon of interest. In the univariate case y_i represents a particular observation on the variable of interest, and $\mathbf{x}_i$ represents the p-dimensional vector consisting of observations on the predictor variables $(x_1, \ldots, x_p)_i$. The regression problem consists of approximating the data generating function g such that $Y = g(X) + \epsilon$, using the available set of observations, where ϵ denotes random disturbances.

The above formulation is general, i.e., not specific to economics or finance. The types of data and variables characterize *economic* regression problems. Economic data can be categorized into three types: time series data, cross-sectional data, and longitudinal data. The characterisation is based on how the data have been collected. Armstrong [Arm78] characterizes the various data types as follows.

- *Time series* data take a given decision unit (e.g., US inflation) and examine it at different points in time. Macroeconomic time series data often have the following characteristics which can cause problems in regression modelling. There are few observations, a lack of variation in the data, substantial measurement errors, interaction, autocorrelation, and multicollinearity. Macroeconomic time series generally provide cheap, fast, and realistic information. In the financial area high frequency data become readily available, such as

hourly or per-minute data on exchange rates.

- *Cross-sectional* data take a given period of time (e.g. January 1993) and examine differences among decision units, such as US households, for instance. The advantages of cross-sectional data over time series data are: larger variations can generally be found in the dependent variable and in the causal factors, there is less multicollinearity, and there is a greater independence among observations. The disadvantage is the loss in realism for situations involving predictions of the future.

- *Longitudinal* data take a sample of decision units (e.g., the interest rate of different countries) and examine changes over time. Longitudinal data generally cost more to obtain, but they provide one advantage over time series data: each observation serves as its own control. In other words, the unique aspects of each decision unit are assumed to be constant over time and are therefore less likely to enter into an explanation about changes.

In this thesis we focus on regression models which use time series and cross-sectional data.

Granger [Gra94] characterises economic time series as having relatively short-length, high levels of measurement errors, nonstationarity, and nonlinear and stochastic relationships. Major macroeconomic series often consist of only 200 to 500 observations as there are, at most, about 40 years of monthly data available since the end of the second World War. As economics is a nonexperimental science, data cannot be just generated. Economic data from further back into history are often irrelevant since the structure and composition of the economy evolves. "The signal-to-noise ratio may well be 3:1 for important macroseries" [Gra94], which implies that measured variables can show behaviour different from the true economic concepts. Several causes come to mind. In a complex economy many economic variables usually are estimated instead of measured directly, which inevitably results in measurement errors. Many variables are also difficult to define, and there is always an economy hidden for official authorities. All these measurement errors become embedded into the economic variables and are not simply added to the signal. "There seems to be no reason to believe that the noise is Gaussian independent and identically distributed" [Gra94]. Frequently, there is a clear trend in mean and in variance of a series, and there is seasonality in mean and possibly in variance; this is roughly what is meant by *nonstationarity*. Nonstationary time series sometimes have a severe impact on the modelling process. In addition, the statistics of economic data lag behind the actual events. If in February forecasts have to be made for a particular macroeconomic variable, the latest available explanatory variables relate, for instance, to the third quarter of the preceding year. If these lags, moreover, differ from one variable to another, the data set is said to have a "ragged edge". This causes problems in real-life forecasting situations. In most theoretical studies this problem is avoided by letting the data collection end at that point

in time from which on the data for a particular explanatory variable are not available anymore. Another data related issue is the reliability of economic data, which can be "measured" by the frequency with which preliminary data are amended, or by the frequency of changes in the definition of a given data series. Finally, there is the discrepancy between the economic concepts and the statistical measurements available. In the absence of adequate direct measurements, indirect measurements or 'proxy' variables have to be used. Different proxy variables for a single economic concept often show different behaviour over time, which clearly hampers the construction of good forecasting models.

The econometric approach to regression modelling consists of three parts: model specification, model estimation, and model evaluation. Economic theory proposes relevant variables for predicting a specific economic phenomenon. The applied economist assumes that the functional form of the predictive relationship has a specific parametric appearance, usually linear. Then the model is estimated, tested and refined, and then re-estimated. When the researcher has found a satisfactory model, he has to evaluate its qualities. This can be done, for example, by comparing his model's predictions with predictions obtained by other models.

The purpose of regression modelling can be *explanation* or *prediction*. A regression model can, for instance, be constructed in order to explain how the dollar-deutsch mark exchange rate depended on a certain set of economic variables in a particular period in the past. A regression model can also be constructed to predict next month's exchange rate given projections of the relevant economic variables. Economic models designed to predict future events do not always explain past events well, and vice-versa. Therefore, the researcher has to determine his main modelling objective: explanation or prediction. In this thesis, however, economic models are constructed solely for the purpose of prediction, mainly because of the important role prediction plays in our society: "Forecasting is a very serious activity in economics, involving a great deal of effort and money to produce them, ..." [Gra94].

Economic forecasts are required for several reasons. The use of forecasting in economic policy-making is a major one; the future is uncertain and the full impact of many decisions taken now, is not felt until later. Consequently, accurate predictions of the future improve the efficiency of the decision-making process [HPT90]. Wallis [Wal89] states: "Outside the policy-making context, two further quite different motives for forecasting exist. One is to anticipate events, whether for private gain or for public good, and it is largely in respect of the former that the growth in forecasting activity has occurred; the other is to put hypotheses about the behaviour of the world to test ...". Here we assume that the main motivation for making predictions is to anticipate economic events, such as a rise or fall in the dollar-yen exchange rate. The predictions could eventually be used for private gain when the information is used in a trading system, for example.

The key ingredients of a forecast still are quantitative data and a framework for their

interpretation and analysis. Nevertheless, substantial developments have occurred in respect of the methods of analysis. Early forecasts were based on a limited number of variables, which were analysed in the context of an implicit, informal model, not necessarily written down. The process relied on the assessment of data and the evaluation of new information by the experienced forecaster. In the 1960s the use of explicit formal models based on estimated equations increased. Nowadays (complex) quantitative models have become generally accepted.

The models distinguish between endogenous and exogenous variables, that is, those determined by the system of equations and those treated as being determined outside the system. In a forecasting context, exogenous variables have to be set by projection or assumption, which leads to further distinction between *unconditional* and *conditional* forecasts. The former represents the conventional understanding of a forecast, namely a prediction of a future event, whereas the latter represents an if-then statement, resting on the occurrence of certain specified conditions [Wal89]. The unconditional prediction is often referred to as *forecast*, while the conditional prediction is often referred to as *prediction*.

The conditional prediction problem is addressed in this thesis. This preference is shared by other researchers; Granger [Gra94], for example, states: " My personal beliefs, which I think are widely shared by other econometricians, are that forecasts derived from relationships between several variables are better than from univariate models...". Univariate time series models, on the other hand, provide us with forecasts purely based on the history of the economic variable of interest. The proponents assume that all possible information is present in the latest value of the economic variable of interest. Univariate time series models do not use any economic theory, which makes them unfavoured among economists.

Predicting the value of Y conditional on the value of X requires both an accurate approximation of the relationship between Y and X and a sufficiently accurate prediction of the value of X. Research usually concentrates only on the first requirement. Since these studies typically use data from past periods, actual values for the exogenous variables are available, which are used instead of predicted X values. These studies, therefore, implicitly assume that the exogenous variables can be accurately predicted. However, the X values have to be predicted, usually by some univariate time series model, and therefore contain prediction error.

In case regression analysis is employed to obtain a rule for predicting future values of the response variable Y given a particular realisation x of X, prediction accuracy is the only important virtue of the model. If, however, the experimenter wants to try to understand the properties of the data generating mechanism g, the interpretability of the approximation f of the model g is also important. Depending on the application rapid computability and smoothness of f are sometimes desirable properties as well [Fri91]. The reader should bear this in mind while reading this thesis which heavily uses prediction accuracy.

Developments in computer software and hardware have resulted in an increase of the size

of the models, i.e., the number of equations and variables. Computer capability is no longer a bottleneck, neither on the size and complexity of the model to be constructed, nor on the econometric estimation and testing procedures applied to the models. The econometric procedures themselves have been substantially developed over the last twenty years. The increment in computer facilities has stimulated research into directions that were previously dismissed as being practical infeasible. Most researchers or financial analysts now have enough computer facilities at their disposal to employ the latest methods and techniques.

The latest relaxation of computing constraints allows for the application of techniques that are able to *extract*, or *learn*, relationships from available data. In contrast with the standard econometric approach the form of the relationship no longer needs to be prespecified by the researcher, but can be shaped by the data themselves. Investigating the employability and relevance of these learning methods for economic prediction problems is interesting, since more accurate predictions, as well as more insight into the underlying economic relationship, can be a consequence. Further, these learning methods provide benchmarks against which the traditional parametric models can be compared and consequently be improved. One critical note, however, has to be made. Despite the vanishing computing constraint, we should remember that one important cause of many problems arising in economic prediction still remains: the limitations of the data. The success of prediction methods is partly determined by the basic properties of the series to which they are applied.

Researchers from different disciplines have been developing methods that are able to perform some form of 'learning', and which go by different names, such as nonparametric regression techniques, semiparametric regression techniques, flexible regression techniques, and learning techniques. Cheaper and more readily available computer power stimulates research in these learning techniques, and makes them available to financial analysts or practitioners of econometrics.

During the last few years one particular learning method has received great attention in literature, namely the *neural network*, also known as the connectionist model. The interest is so overwhelming that scientists already call it a hype. The occasional interest of popular journals in this method confirms this statement. The literature reports many successful applications of neural networks to financial problems. Although opponents, mainly statisticians, are not convinced of their capabilities, it cannot be denied that the attention neural networks receive from researchers from different disciplines makes them develop fast in contrast with their statistical competitors.

The financial applications, however, are mainly concerned with univariate time series modelling (such as forecasting the dollar-yen exchange rate on a day-to-day basis) and classification (such as bankruptcy prediction). These studies merely intend to illustrate how well neural networks perform when compared to traditionally used techniques. Although such research cer-

tainly stimulates the "marketing" of neural networks, there is a growing need for more detailed knowledge on methodological aspects of applying neural networks. Additionally there is the need for economic specific knowledge. In particular, which specific characteristics of economic time series influence the applicability of neural networks and in what sense?

Applying neural networks to financial (time series) modelling problems is the subject of this thesis. *The aim of this thesis is to gain insight into the employability and practical relevance of feed-forward neural networks for the specification of multivariate economic (time series) models that are used for the purpose of prediction.* Neural networks are regarded as practically relevant when they are able to improve upon the prediction accuracy of traditional approaches in practical situations. The foregoing indicates that in this research the three key concepts are: prediction, economic time series, and neural networks. The research concentrates on the conjunction of these three concepts. The relevance of this research to economics is indicated by [Hal94]

> The traditional neural network literature has offered little firm guidance in the way the specification choice can be made.
> ... The combination of the extremely rich functional forms of the neural network and the tools of statistical inference offers a potentially promising and exciting new avenue of research to the forecasters, although much remains to be done to prove the practical usefulness of these techniques, especially for small-sample applications.

Research activities are subdivided into three groups: relevant theory selection and discussion, neural network strategy development, empirical research using Monte Carlo experiments and case studies. Theoretical aspects of multivariate time series modelling, nonparametric regression, neural network learning, and comparing prediction accuracy will be discussed. Since, until today, no clear methodology or strategy is available to the neural network practitioner, this research requires a clear description of the strategy followed. Monte Carlo experiments are performed to illustrate some practical aspects when applying neural networks, such as overfitting, local minima, and the effectiveness of weight decay. Monte Carlo experiments enable the researcher to perform controlled experiments, which are otherwise impossible in economic research. It should be noted, however, that the relationships revealed are conditional on the design of the Monte Carlo study, which will not exactly represent the situation as encountered in practice. Consequently, the real *practical* relevance of neural networks is assessed in three economic prediction problems, viz., the prediction of hedonic house prices, the prediction of new mortgage loans, and foreign-exchange rate prediction.

The thesis consists of two parts. Part I, which includes the Chapters 1, 2, 3, 4, and 5, discusses the theoretical aspects of economic modelling in general and with neural networks. Part II, which includes the Chapters 6, 7, 8, 9, and 10, deals with the practical aspects in applying neural networks.

Chapter 1 summarises the most important aspects of economic (regression) modelling. The modelling process is divided into three parts: model specification, model estimation, and model evaluation. The aspects of each part, which are relevant for the line of reasoning in this thesis, are discussed. One important aspect of model specification is the functional form of the model. In econometrics, the functional form is specified by the investigator himself, usually based on pragmatics. Linear models are most frequently used in applied work. Economic theory normally does not give much information on the functional form of the model, but sometimes the data themselves do. This may be a reason why nonparametric techniques have potential for financial problems. We intend to make neural network researchers who are *technique* dedicated aware of the accumulated knowledge that exists for the problem domain they apply their technique to.

Chapter 2 summarises several model free regression methods, among them neural networks. The concepts of the different approaches are briefly discussed, as well as the position of the neural network among the others. A problem of great concern to nonparametric regression is ending up with a model that is (almost) unbiased, but which has high variance. This problem, which affects *all* members from the class of nonparametric methods, is known as the bias/variance dilemma. Most nonparametric methods have parameters that determine the flexibility of the resulting fit. Cross-validation is a method designed to make a good parameter choice. Finally, the procedure of cross-validatory parameter choice is described.

Chapter 3 discusses the theoretical aspects of neural networks, such as the mathematical representation and learning theory from a statistical perspective. To assess the practical relevance of neural network models, the accuracy of the predictions is often compared with the predictions made by traditional models. Chapter 3 provides some practical methods for making statistically sound comparisons among different (economic) prediction models, and discusses the relevant statistical issues.

When applying neural networks, one probably meets difficulties that determine the practical success, such as how many hidden units to take, how to deal with local optima, and how to reduce overfitting. These and many other difficulties that are typically met in practice will be discussed in Chapter 4. Based on this discussion, a neural network strategy is determined, which will be used in all experiments.

Time series, especially nonstationary, impute many difficulties in the econometric modelling process. Therefore, the econometric analysis of time series requires additional attention. Chapter 5 discusses modern issues in this area, such as testing for unit roots, cointegration, and error-correction. These concepts originate from a linear viewpoint of constructing models. Chapter 5 is a first step towards a nonlinear generalisation of these concepts, using neural networks.

Chapter 6 reviews a sample of the literature on neural network applications to economic and financial problems.

In Chapters 7, 8, and 9 we apply neural networks to the prediction of the hedonic price of housing in Boston, the prediction of the production of new mortgage loans in the Netherlands, and the prediction of foreign exchange rates, respectively. These three case studies show how neural networks are applied in economic practice, and how accurate their predictions are compared to simple linear models for *real* financial problems.

Chapter 10 gives a summary, draws conclusions from the research undertaken, and suggests some directions for future research.

Part I

Theory

Chapter 1

The Economic Modelling Process

1.1. Introduction

This chapter addresses several important issues in the process of economic (timeseries) modelling. Econometrics has been performing research in the quantitative aspects of this problem for decades. "Econometrics is the field of economics that concerns itself with the application of mathematical statistics and the tools of statistical inference to the empirical measurement of relationships postulated by economic theory." [Gre93]. Assuming a 'convenient' probability structure for the model makes it possible to deduce the properties of estimators and test statistics to assess their value. The test statistics may then be used to confront specific economic hypotheses with the empirical evidence presented by the data. More generally, the test statistics provide a guide to choosing between the different specifications suggested by economic theory.

Applied work in economics often presumes linear models. The reason is that econometric theory is best developed for this particular class of functions. Many statistical inference methods are at the investigator's disposal. In this chapter we pay attention to modelling within the class of linear models. Nowadays nonlinear modelling is becoming more and more popular by both theorists and practitioners; see the next chapters.

Two main objectives of econometrics are: *explaining* the behaviour of economic entities in the past, and *predicting* the behaviour of economic entities in the future. A single model need not meet both objectives simultaneously. For instance, a model good at predicting future behaviour can offer little or no insight into the underlying relationships; on the other hand, a model that satisfactorily explains past behaviour may predict badly.

Three main aspects in modelling are *model specification*, *model estimation*, and *model evaluation*. Model specification concerns the specification of an empirical model that reflects an economic theory or some practical experience. Model specification has been -and still is- the subject of many debates among econometricians (see [Gra90, Keu94]). An empirical model

can be misspecified in at least three ways. First, the set of variables included can be incorrect; irrelevant variables are included, relevant variables are omitted, or both. Second, in a time series context the dynamic structure of the equation can be incorrect. Third, the functional form deviates from the one specified in advance. Model estimation concerns the statistical estimation of the free parameters in the model, using the sample data. When enough model assumptions are made, statistical inference can take place. Textbooks on econometrics in general pay more attention to model estimation than to model specification, although the latter probably is of greatest fundamental importance. Once a model has been estimated, others than the builder himself will often use it. These users have to be convinced of particular qualities of the model. In the model evaluation phase the model builder evaluates his model, usually by performing diagnostic tests and by comparing its performance with the performance of alternative models.

The basic techniques of estimation, testing, and specification are applicable to both cross-section and time series problems. The element of time, however, adds a new dimension to economic modelling. It raises important questions concerning the interpretation of a model, particularly with respect to equilibrium and steady-state growth, and it brings in a whole range of statistical considerations concerned with modelling variables which do not adjust instantaneously to changes in other variables.

In our conception time series are stochastic processes $\{X_t | t \in T\}$, i.e., a set of real valued random variables which are indexed by t, where t represents time. *Stationary* and *nonstationary* time series constitute two important classes of time series. A stochastic process is said to be stationary in the *strong sense*, if the joint probability distribution of the n realisations of the process is time independent (see [BDGH93, page 11] for a detailed definition). A more practical variant is weak stationarity. A stochastic process is said to be stationary in the *weak sense* (or stationary for short) [BDGH93, Har90] if :

$$E(X_t) = E(X_{t+h}) = \mu < \infty$$
$$\text{Var}(X_t) = \text{Var}(X_{t+h}) = \sigma^2 < \infty$$
$$\text{Cov}(X_t, X_{t+j}) = \sigma_j < \infty$$

If at least one of the conditions above is not fulfilled, the process is said to be *nonstationary*. Many macroeconomic time series appear to be nonstationary. Throughout this chapter we presume stationary time series; otherwise, it will be stated explicitly. An in-depth discussion of econometric modelling with nonstationary time series will be given in Chapter 5.

Some parts of the remainder deal with time series data exclusively; some parts are for both time series data and cross-sectional data. The context indicates which case we have; otherwise, it will be stated explicitly.

The outline of the chapter is as follows. Section 2 discusses the model specification part

of the economic modelling process. Section 3 addresses several issues in model estimation. Section 4 deals with model evaluation. Section 5 concludes the chapter.

1.2. Model Specification

1.2.1. The 'Textbook' Approach

In the 1940s the emphasis in econometric practice moved from measurement of parameters and quality of statistical data to testing of economic theories. The methodology which reflects this emphasis is now identified with the Cowles Commission. At that time there was no formal methodology as to how empirical work should be performed. The assumptions of the Cowles Commission methodology, which were often implicitly present in empirical work, are summarised in [CD92]. The main assumption is that one knows the correct specification; this assumption, however, is unacceptable in a nonexperimental domain [Lea78]. Practice subscribes to this viewpoint; instead of finding one model for a single phenomenon, a whole series of models are found all explaining or predicting the same phenomenon. For instance, several alternative models of exchange rate determination can be found in the literature [MT92].

According to [Gra94] a simplistic form of the classical modelling procedure –often referred to as the 'textbook' approach– starts with an economic theory, restates this in the form of estimable equations, and finally estimates the parameters. Some simple evaluation information is given, such as R^2, t-statistics, and possibly the Durbin-Watson statistic. Finally, the resulting model is interpreted and its 'policy implications' are explored. Anyone trying to follow this sequence is likely to meet several problems. Hence, many decisions concerning model specification have to be made in practice, e.g., which variables to include in the model, whether to model the equilibrium or the disequilibrium situation, and what functional form of the relationship to use.

In the 1970s there was a growing scepticism towards the value of traditional econometric analysis. Econometric research in the 50s and 60s, which had been full of optimism, had not addressed the real practical problems of model specification and selection [CD92]. The applied researchers developed their own methodological approach. When computing power became more readily available, routine calculations of statistical tests could be easily carried out, which resulted in the following approach. Economic theory is used to specify the appropriate variables in the empirical regression equation. This equation is then estimated, and assessed using the normal t-statistics, R^2, and the usual tests for autocorrelation and heteroscedasticity. The response to unsatisfactory test results (e.g., insignificant variables) is to modify or 'improve' the equation in some way (e.g., by leaving out those variables that are insignificant). At the end the final model is interpreted as though it was the first and only equation tried. This procedure, however, make its value questionable at least. The procedure is an example of what is commonly

referred to as *data mining*. In general, given a limited amount of data and a huge number of possible models, there is always a possibility that, if enough models are fitted to the data, one will appear to fit the data very well. In fact this good fit may be due to chance alone, and the corresponding model will then be useless.

Specification search is proposed by Leamer [Lea78] as a positive identification of the issue of data mining, which is negatively loaded in econometrics. In the Machine Learning community, however, a new branch with the name data mining (or data base mining) is developing. In this context data mining, or knowledge discovery in (large) databases, uses a set of inductive learning techniques to extract knowledge from databases. The two should not be confused. We will therefore use the term specification searches for this thesis.

1.2.2. Specification Searches

The theoretical 'textbook' approach neglected model specification as part of the methodology for empirical modelling. In practice, as we have stated, applied economists actually did search –and still do– for a model specification that suits both the economic theory and the actual data. Researchers are driven by various motives when searching for a suitable model specification. Leamer [Lea78] discerns six different types of specification searches, which are presented in Table 1.1. A hypothesis-testing search tests a specific hypothesis about the phenomenon, e.g.,

Table 1.1: Different types of specification searches (from [Lea78])

Name of Search	Designed to
Hypothesis-testing search	choose a "true" model
Data-selection search	select a data set
Interpretative search	interpret multidimensional evidence
Proxy search	find a quantitative facsimile
Simplification search	construct a "fruitful" model
Post-data model construction	improve an existing model

the dollar-deutschmark exchange rate is independent of the mutual interest rate differential. In a data-selection search two identical model specifications differ in their choice of data sets; the data set which results in a model with favourable test statistics is selected. In a proxy variable search the best measurement of a hypothetical variable is selected, e.g., which measurement of inflation results in a good model fit. In an interpretative search, the underlying hypothesis is taken as given, and restrictions are imposed on the parameters in the hope that the estimates may be "improved". The process of revising the underlying theory in response to the data evidence

is called post data model construction. The final search is the simplification search, in which one tries to find a simple but useful model, i.e., a more parsimonious model which predicts the future as well as the more complex initial model. Leamer [Lea78] recognises that in practice it can be difficult to infer what kind of search actually occurred, since often the searches differ only in the intent of the researcher and not in his actions.

1.2.3. The General to Specific Approach

The recognition of specification uncertainty has inspired methodological responses to specification searches; the best-known are the ones initiated by Hendry, Sims, and Leamer, respectively (see the collection of articles [Gra90]). All three methodologies start with a general initial specification and try to reduce its complexity in various steps. Both the Hendry and Sims methodologies are specific to time series and presume linear models. We will discuss only the Hendry approach, since it is most frequently followed in empirical work.

The Hendry approach is generally known as the general to specific approach. The software package *PcGive* has been designed according to the philosophy of this approach. The approach essentially comprises four steps ([Pag87, Har90]):

1. Formulate a general model that is consistent with what economic theory postulates about the variables entering any equilibrium relationship and which restricts the dynamics of the process as little as possible. This model provides a yardstick against which the more restricted models may be assessed.
2. Reparameterise the model to obtain explanatory variables that are nearly orthogonal and which are 'interpretable' in terms of the final equilibrium. In many cases this reparameterisation means formulation of an error-correction model (ECM), to be discussed later.
3. Simplify the model to the smallest version that is compatible with the data ('congruent'), using any prior economic theory to suggest a suitable specification for the dynamics.
4. Evaluate the resulting model by extensive analysis of the residuals and predictive performance, aiming to find the weaknesses of the model designed in the previous step.

The Autoregressive Distributed Lag (ADL(p, q)) model

$$y_t = \sum_{i=1}^{p} \alpha_i y_{t-i} + \sum_{i=0}^{q} \boldsymbol{\beta}_i^T \mathbf{x}_{t-i} + \epsilon_t$$

is usually taken as initial model. It subsumes nine different types of models by putting certain restrictions on the parameters ([Hen93, p.447-454]), which makes it a good general model to start with.

1.3. Model Estimation

When a model has been specified, the free parameters have to be estimated so that the model fits well to the data at hand. The starting point is the classical linear multiple regression model

$$Y_i = \beta_{1i} + \beta_{2i} X_{2i} + \ldots \beta_{ki} X_{ki} + \epsilon_i \quad i = 1, \ldots, n, \tag{1.1}$$

which is completed by the following assumptions:

1. The explanatory variables $\mathbf{X}$ are (a) nonstochastic, (b) have values fixed in repeated samples, and (c) are such that $(1/n)\mathbf{X}'\mathbf{X} \rightarrow \mathbf{Q}$ where $\mathbf{Q}$ is a nonsingular matrix of finite constants.
2. The rank of $\mathbf{X}$ is k.
3. $E(\epsilon_i) = 0$ for all i.
4. $\text{Var}(\epsilon_i) = \sigma^2 = \text{constant}$ for all i.
5. $\text{Cov}(\epsilon_i, \epsilon_j) = 0$ for all $i \neq j$.
6. The number of observations exceeds the number of parameters.

Provided that these assumptions hold, the OLS estimators $\widehat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'Y$ possess all desirable large and small sample properties of unbiasedness, efficiency, and best linear unbiasedness (BLUE). Most econometric textbooks (e.g. [JG85, WW79]) pay much attention to estimation techniques, in particular to the difficulties that arise when some of the assumptions of the classical regression model break down, such as autocorrelated disturbances, heteroscedasticity of the disturbances, multicollinearity of the explanatory variables, and so forth.

Non-randomness of the explanatory variables is obviously an implausible assumption for virtually all economic data. In economics the investigator almost always has to accept whatever data observations are available, rarely being able to fix the values of any of the variables in which he is interested; the economic system under observation not only determines the disturbances, but also the values of the explanatory variables. In this case the OLS estimators retain the properties of unbiasedness and consistency only when each and every explanatory variable or regressor is independent of all disturbance values, past, present and future.

1.3.1. Measurement Errors

Many economic data series only approximate the "true" underlying values of the variable that the investigator wants to measure. For example, many economic variables are approximated on the basis of a sample, typing errors are frequently being made, data series measure concepts that sometimes differ from those that appear in economic theory, and there often is a lack of proper updates which makes extrapolation necessary. It seems likely that these measurement errors

occur randomly and only in some parts of the data series. Therefore, we assume the measurement errors to be Gaussian distributed with zero mean and constant variance. We consider the two variable regression case as an example. Suppose the true regression is represented by

$$Y_t = \alpha + \beta X_t + \epsilon_t, \quad t = 1, 2, \ldots, n \tag{1.2}$$

where ϵ_t is i.i.d. $(0, \sigma^2)$. Suppose, however, that instead of (Y_t, X_t) we observe (Y_t^*, X_t^*) where $Y_t^* = Y_t + \nu_t$ and $X_t^* = X_t + \omega_t$ and ν_t and ω_t represent the measurement errors in Y_t and X_t respectively. We attempt to estimate the parameters (α, β) from the observed values rather than from the true values. The original regression equation (1.2) can be written as

$$Y_t^* = \alpha + \beta X_t^* + (\epsilon_t - \beta \omega_t + \nu_t), \quad t = 1, 2, \ldots, n. \tag{1.3}$$

The problem is that the composite disturbance term is correlated with the independent variable, which makes OLS a biased and inconsistent estimator of the parameters (α, β).

1.3.2. Multicollinearity

A set of regressors is said to be completely multicollinear when there exists at least one regressor that is a linear combination of the others. This regressor adds to the dimension of the variable space, but it does not provide enough information to model the relationship in this extra dimension. Figure 1.1 illustrates what happens if two regressor variables X and Z are

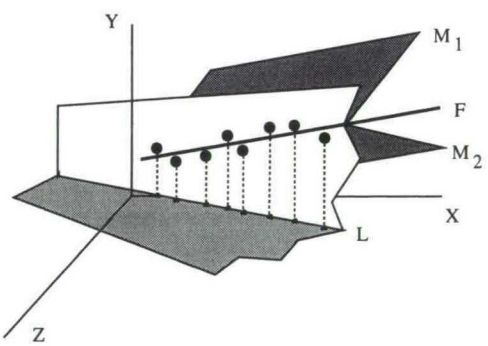

Figure 1.1: Multicollinearity (taken from [WW79])

collinear. In this case the projection of all observations onto the XZ-surface are restricted to the line L. This means that in fact we have information only on a thin slice of the relationship. The observations do not say anything about the shape of the relationship in XZ-directions orthogonal to L; an infinite number of surfaces fits through F (for example, $M1$ and $M2$).

In [JG85] the following consequences of multicollinearity are given:

1. It becomes very difficult to precisely identify the separate effects of the variables involved.
2. Unknown parameters may not appear significantly different from zero, and consequently variables may be dropped from the analysis, not because they have no effect, but simply because the sample is inadequate to isolate the effect precisely.
3. Estimators may be very sensitive to the addition or deletion of a few observations or the deletion of an apparently insignificant variable.
4. Despite the difficulties, accurate forecasts may still be possible. This is only true, however, if the pattern of interrelationships among the explanatory variables is the same in the forecast period as in the sample period.

Multicollinearity does not change the theoretical properties of OLS, such as unbiasedness and BLUEness. The only impact multicollinear regressor variables have on OLS is that the variances of the parameter estimates become large, such that confidence intervals become so wide they do not provide useful information anymore. The highly variable parameter estimates consequently result in very wide prediction intervals.

Possible approaches to reduce the variances in the parameter estimates are leaving out highly correlated regressor variables or employing, so called, statistical shrinkage methods. Shrinkage methods, such as, ridge regression, principal components regression, and partial least squares [FF93], shrink their parameter estimates away from low-spread directions in the regressor space. This mainly serves to reduce the variances of their estimates, and this is what gives them generally performance superior to OLS estimation [FF93]. Reducing the variance in the estimates effectively improves prediction performance [MF].

Subset selection constitutes an alternative strategy to decrease the variances in the parameter estimates. From the total set of variables which are initially thought to affect the variable of interest, a smaller subset is selected. Hence, for linear models fitted with OLS on a relatively small set of observations, the more variables are added to a model the larger is the variance of the predicted values [Mil90]. The increase in variance must be traded against the decrease in bias. To control the variance in the predictions, it is better to look for a model that consists of a subset of the total set of predictive variables. Subset selection must be carried out carefully, taking into account the bias in the parameter estimates that arises when the same set of observations is used for selecting the best subset of variables and for the estimation of the parameters in the selected model [Mil90].

1.3.3. Heteroscedasticity

Consider the scatterplot shown in Figure 1.2 where the true relation is indicated by the dashed line. The vertical spread of the observed values from the true regression line increases as the regressor X increases. We speak of heteroscedasticity when the variance of the disturbance term is not constant. The observations on the right in Figure 1.2 give a less precise indication of where the true regression line lies than the observations on the left. In this case OLS remains unbiased, but no longer BLUE. The sampling variances of the OLS estimators, however, are biased estimates of the true values, which means that we can no longer rely on the usual inferential procedures (e.g., for hypothesis testing).

It seems reasonable to pay less attention to the observations on the right than to the more precise observations on the left. This is the philosophy underlying weighted least squares (WLS), which is favoured above ordinary least squares (OLS) for such situations. Instead of using WLS, it is sometimes possible to transform the endogenous variables (e.g., by a log transformation) and to apply OLS.

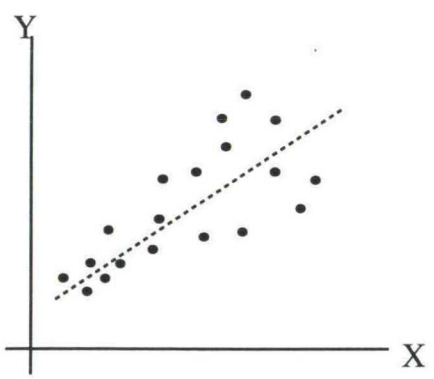

Figure 1.2: Linear regression scatterplot when σ_i is proportional to X_i

1.3.4. Serial Correlation

A key assumption in the classical linear regression model is that the disturbances ϵ_t and ϵ_t' ($t' \neq t$) are uncorrelated. Autocorrelation, which in a time series context is usually called serial correlation, means that successive observations are dependent to some extent. In case of positive autocorrelation, the second (or some later) observation tends to resemble the previous one, and hence gives less information about the relationship than independent observations would give. When dealing with time series the assumption of uncorrelated disturbances may

not always be reasonable. Hence, the disturbance term can be regarded as being made up of a number of omitted variables, which in a time series context will likely show some degree of serial correlation since relatively few time series are random.

For small samples serial correlation has two consequences [WW79]. First, the serially correlated disturbance terms make estimation of the regression parameters erroneous. This effect reduces with increasing sample size. Second, the observed error terms (residuals) show less variance than the true error terms, which makes confidence intervals for the parameters erroneous; hence, the estimators of the standard errors are biased.

In general, applying OLS to a model with serially autocorrelated disturbance terms still yields unbiased parameter estimates, but these estimates have larger variances. As a consequence, we can no longer rely on the standard testing procedures.

The traditional method of handling serially correlated disturbances is to model these disturbances by an AR(1) process,

$$\epsilon_t = \phi\,\epsilon_{t-1} + \nu_t, \quad |\phi| < 1,$$

although with modern computer technology any ARMA(p, q) model can be chosen. The coefficients of the ARMA process of the disturbances are estimated and the observations are transformed such that the resulting equation meets the standard assumptions of the linear regression model. In many textbooks (e.g. [Har90, WW79, JG85]) the topic of estimating models with serially autocorrelated disturbances is discussed profoundly.

Monte Carlo experiments according to the same design as employed in [BM78, Har90], indicate that in the presence of AR(1) disturbances the variances of the OLS-estimates increase more when X_t is a trending series than when X_t is a random stationary series. Intuitively this makes sense; although subsequent disturbances may look alike, they are dispersed by the X-values, so they cannot easily 'pull' the estimated regression out of position. The experiments further show that disturbances which contain a large autoregressive coefficient ϕ enlarge the variances of the OLS-estimates more than disturbances which contain a small autoregressive component.

1.4. Model Evaluation

When a regression equation has been specified and its parameters have been estimated, the next step is to assess its quality for the purpose it is designed for. The evaluation should be sufficiently detailed, and should be constructed to convince the user (of the model) that the model is suitable for the problem at hand. It seems reasonable to state that the process by which the model is specified determines the extensiveness of its required evaluation. Hence, a user will need more 'evidence' to change his belief when a model is found after an extensive specification search than when a model is directly derived from theory.

Granger [Gra94] mentions some of the existing controversies in model evaluation. Should an 'out-of-sample' evaluation be used, such as a forecasting comparison or a cross-validation exercise? Should a battery of diagnostic tests (to be defined later) be used to check the specification and to suggest respecification? Should the modeller specify his own alternative models or use those of others in a comparison exercise? Statistical testing seems to be the evaluation method most extensively relied on by applied econometricians; in [KM94] it is stated "Testing hypotheses belongs to the basic pastimes of econometricians...A casual investigation of titles of papers [in economic journals] show that there is a lot of 'testing' in the literature".

The applied econometrician uses a whole battery of tests throughout an empirical study. In [KM94] four distinct aims of testing are discussed: theory testing, validity testing, simplification testing, and testing for making decisions. Theory testing is the most ambitious one; it is an attempt to meet the requirements of real science: confronting theory with facts. Validity tests are performed in order to find out whether the statistical assumptions underlying some model are credible. The value of this test should be interpreted with care; hence, a very neat 'valid' statistical model may be obtained after extensive manipulation with the data. It should be noted that the significance levels (the probability of making a type I error) become inflated by extensive data mining. It is recognised by many authors, but not often met in their works, that the real test of a theory is its predictive ability on an independent set of data [KM94]. This is known as out-of-sample evaluation. A model obtained after extensive unacceptable data mining may display bad out-of-sample behaviour. Out-of-sample evaluation is rather costly with respect to the data, since a subset of the total data sample has to be put apart and can therefore not be used for estimation.

When a researcher has to choose among competing models and the choice cannot be made on the basis of statistical tests, Harvey [Har90, p.6-7] and Hendry [Hen93, p. 412-414] suggest the following criteria:

Parsimony Models should concentrate on the most relevant and important aspects of the DGP (data generating process) and consign unimportant aspects to the disturbance term. From the statistical point of view, the key feature of a simple model is that it contains a small number of variables.

Identifiability A model is nonidentifiable if more than one set of parameters is consistent with the data. In practice this means that the estimates cannot be interpreted in any meaningful way. Identifiability is related to parsimony; the more parsimonious a model, the less likely it will suffer from identification problems.

Data coherency A model should approximate the observations in the sample reasonably close, e.g., measured by R^2.

Data admissibility A model should be unable to predict values which violate definitional constraints. For example, an interest rate can be positive only.

Theoretical consistency A good model should be consistent with what is known a priori.

Predictive power A good model should provide accurate predictions of future (out-of-sample) observations. For a model to do this, its parameters must obviously remain constant over time. A clear distinction should be made between the R^2 in the sample and the goodness-of-fit in a post-sample, which is the real test of a model.

Encompassing A model encompasses a rival model if it can explain the results given by the rival formulation; in that case, the rival model contains no information which could be used to improve the preferred model.

Weakly exogenous regressors Explanatory variables should not be contemporaneously correlated with the disturbance term. A major cause of contemporaneous correlation is the simultaneity of many economic relationships. The regressors and the dependent variable may be jointly determined by the simultaneous system in which the equation of interest is embedded.

1.5. Conclusions

A general framework for economic modelling was presented above, which forms the point of departure for the subsequent chapters. The economic modelling process was partitioned into three phases: model specification, model estimation, and model evaluation. These phases were discussed separately, although in practice they often interfere. The main issues in each phase were addressed. This helps in better understanding the role and positioning of Artificial Intelligence (AI)-techniques in the general process of economic modelling, in which traditionally the quantitative aspects are approached with econometric techniques.

The methodological aspects of the model specification phase deserve much attention. A core concept is data mining or specification searches, which entails the following. In general, an economic theory is equally well supported by several different empirical models. The investigator manually searches for an empirical model that fits the data well with respect to some statistical criteria. However, presenting only the final model and the corresponding statistics (used during the search process) invalidates a standard interpretation of the model. It is essential to report the extensiveness of the specification search process to the user, and to provide him with model evaluation measures *not* used in the search process. The general to specific approach is an attempt to structure and control the specification search process and to make valid interpretations of the final model(s).

Model estimation is discussed in many econometric textbooks. The main issue is to derive (asymptotic) distribution theories in case one or more assumptions made in the classical linear regression model are not fulfilled. The statistical shrinkage methods, which we presented as a response to multicollinear data, provide biased parameter estimates and are, therefore, not popular among econometricians (see [JG85]). In later chapters we will adopt the shrinkage technique for a similar purpose.

The objective of model evaluation is to convince (potential) users of the model of its qualities. Although many statistical tests have been developed for this purpose, assessment of the out-of-sample prediction performance is definitely required, in particular, when weak or heavy data mining is employed in specifying the model.

In the context of econometric time series modelling the specification search process is usually constrained to the class of parametric models, i.e., models of which the functional form is prespecified by the investigator. In the subsequent chapter we will present several methods that automatically search for a good approximation to the data, without presuming a particular functional form. In fact these methods automate part of the specification process which an applied economist (implicitly) performs.

Chapter 2

Flexible Regression

2.1. Introduction

The foregoing chapter introduced the three phases that the model building process consists of, namely, model specification, model estimation, and model evaluation. In the model specification phase the researcher has to specify the functional form of the economic relationship. Sometimes economic theory suggests a specific form, but usually it does not. This chapter discusses methodologies used in statistics and AI that let the data themselves determine the functional form of the relationship. In flexible regression the search for an (economic) model is not constrained to a prespecified parametric class of models, as was the case in the previous chapter. Parametric function means that the functional form of the approximating function is prespecified up to some finite dimensional vector of unknown parameters, which has to be estimated from the data. The methods considered here are known as model free, flexible, or non-parametric methods. All of these methods have originally been designed for general regression problems, using cross-sectional types of data. In this chapter we will outline the general philosophies behind these methodologies; we will skip the tedious details and practical difficulties.

Several techniques have been developed for approximating the underlying regression function $g(\mathbf{x})$, defined as $E[Y|X = \mathbf{x}]$, that is, the conditional expectation of the dependent variable given a particular realisation $\mathbf{x}$ of the vector of independent variables X. Traditionally, most of the research on function approximation in high dimensional spaces is pursued in statistics. The principle approach has been to fit a parametric function to the training data, most often by least-squares. The most commonly used functional form (parameterisation) is the linear function

$$y = g(\mathbf{x}) = \beta_0 + \sum_{i=1}^{p} \beta_i x_i + \epsilon, \quad \epsilon \text{ i.i.d.}(0, \sigma^2). \tag{2.1}$$

The previous chapter concentrated on particular aspects of (linear) modelling in economics that

are frequently encountered in practice. In econometrics the functional form is usually based on economic theories of the particular phenomenon. This knowledge, if present, is usually not accurate enough to specify a functional form which meets the true underlying relationship. The parametric function has limited flexibility and is likely to produce an accurate approximation if the underlying function $g(\mathbf{x})$ is close to the specified parametric one. On the other hand, there are practical advantages, such as the requirement of relatively few observations, the ease of interpretation, the absence of almost any computational effort, and the availability of strong mathematical theories that allow for rigorous analysis. The stochastic disturbance term ϵ in (2.1) captures the influence of all variables omitted from $\mathbf{x}$ and the influence of all irrelevant variables included in $\mathbf{x}$. If the noise term ϵ is large compared to the 'signal' $g(\mathbf{x})$, then the systematic error made by misspecification of the functional form influences predictive accuracy only marginally.

Since computer power continues to increase and to become ever cheaper, it is attractive to enlarge the search space for good models. It becomes feasible to let the data determine the functional form. This sometimes results in highly nonlinear relationships. At the moment these flexible regression techniques, as we call them after [Rip93b], are increasing in popularity; one technique that surpasses all the others with respect to popularity is the neural network method. It is obvious that the more complex a relationship becomes, the more data are needed to approximate it sufficiently accurately. The various methodologies we discuss differ, in the degree of flexibility, in the requirements of data, and in the quality of the resulting approximation.

There is no general agreement on the meaning of the terms parametric, semiparametric, and nonparametric regression. Parametric techniques bias the search to a small set of models. The model is a representation of what the modeller thinks the data generating system actually looks like. In its most ideal form, the parametric model should contain only parameters that have a clear interpretation. Nonparametric –also called model free or flexible– regression techniques, on the other hand, attempt to *'learn'* the model from the data without presuming any functional form. Nearest neighbour regression, which approximates $g(\mathbf{x})$ by averaging over $g(\mathbf{x_i})$ where $\boldsymbol{x_i}$ is in the neighourhood of $\mathbf{x}$, is a typical example of a nonparametric technique. Flexible regression techniques usually need a large amount of data to obtain results that are statistically meaningful. An advantage of these techniques is that the need for imposing a bias on the model is reduced. There are some techniques, such as neural networks (see Chapter 3 and 4), which look like a mix of the two. When a specific architecture is given, the network is parametric, in the sense that one is trying to find the maximum likelihood values for the weights (see the section on neural network learning); but at the same time it is model free, since we do not really believe that the true underlying function is a composition of sigmoids. However, we do know that we can get close to any continuous function [Cyb89, Whi89b]. It is clear that the parameters do not have a theoretical meaning. We prefer to use the term model free or flexible regression

to contrast parametric regression.

In general, model free regression methodologies are "consistent" for essentially any function $E[\mathbf{y}|\mathbf{x}]$. *Consistency* is defined as the asymptotic (large sample) convergence of an estimator to the true $E[\mathbf{y}|\mathbf{x}]$. The consistency feature is seen as a necessary condition for the use of a particular model free technique. However, as we shall see later, consistency in itself does not imply a successful application of the method to a single finite data sample. It has been proved that under certain conditions neural networks are consistent [Cyb89].

The outline of this chapter is as follows. Section 2 summarises the conceptual ideas behind several well-known flexible multivariate regression methods, among which are neural networks. Section 3 introduces *the bias /variance dilemma*, which forms a general problem for all model free regression methods. Section 4 discusses cross-validation, a procedure that is often used to choose flexibility parameters. Section 5 concludes the chapter.

2.2. Flexible Regression Methodologies

Friedman [Fri91] reviews the existing methodology for multivariate regression. Figure 2.1 gives the different model free regression approaches from his review. In the following subsections we shall characterise the methodologies represented in the figure, summarise their strategies, their advantages and drawbacks, and give some representatives.

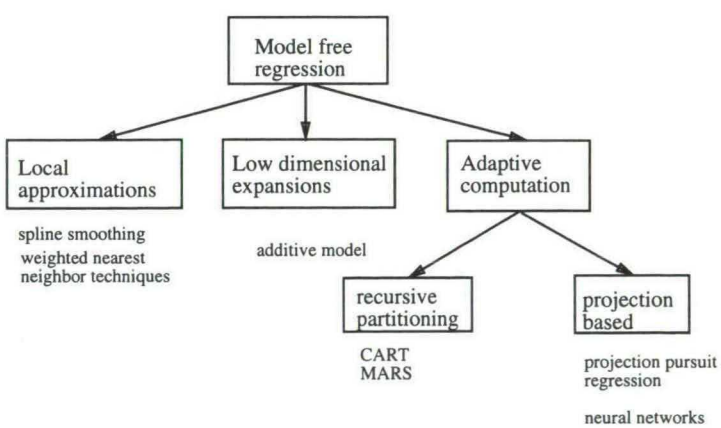

Figure 2.1: Existing Methodology (source: [Fri91]

2.2.1. Local Approximations

Local approximation methods are multivariate extensions of univariate scatterplot smoothing techniques, such as nearest neighbourhood techniques, spline smoothing techniques, and kernel smoothing techniques. See [Hae90, Alt92] for an extensive discussion of univariate scatterplot smoothers. The general underlying principle is local approximation, by locally fitting polynomials or by locally weighted (or unweighted) averaging. Theoretically, the basic idea of scatterplot smoothing can be straightforwardly extended to higher dimensions. However, there are two major problems with this approach in multidimensional feature spaces. First, a geometrical description of the regression relationship between X and Y cannot be provided; its form cannot be displayed for dimensions higher than two. Second, the basic principle –averaging over local neighbourhoods– will often be applied to a relatively limited set of points, since samples of respectable size ($n = 1000$) are surprisingly sparsely distributed in higher dimensional Euclidean spaces. The quality of the approximation of $E[Y|X = \mathbf{x}]$ depends on the number of observations in the neighbourhood of $\mathbf{x}$; averaging over a large number of neighbouring observations evidently gives a more accurate approximation than averaging over a small number of observations. This problem is known as 'the curse of dimensionality' [Bel61].

2.2.2. Low Dimensional Expansions

The ability of local approximations to provide adequate approximations in low dimensions, coupled with their inability to adequately approximate functions in high dimensions, has motivated approximations that take the form of expansions in low dimensional functions ϕ_j

$$f(\mathbf{x}) = \sum_{j=1}^{J} \phi_j(\mathbf{z}_j),$$

where each $\mathbf{z}_j$ is comprised of a small –usually one or two elements– (preselected) subset of $\{x_1, x_2, \ldots, x_p\}$. After selecting the variable subsets $\{\mathbf{z}_j\}_1^J$, the corresponding function estimates $\{\phi_j(\mathbf{z}_j)\}_1^J$ are obtained by some local approximation method in conjunction with the *backfitting algorithm* [HT90, Fri91]. The backfitting algorithm is a general, iterative, algorithm that enables one to fit an additive model using *any* regression-type fitting mechanisms. It considers each variable subset $\mathbf{z}_j$ in turn, and smooths the residuals $(y_i - \sum_{k \neq j} \phi_k(\mathbf{z}_{ik}))$ against the predictor variables $\mathbf{z}_{ij}$ to estimate ϕ_j. The process is continued until convergence. For example, with least squares, the backfitting algorithm iteratively estimates $\phi_j(\mathbf{z}_j)$ by

$$\phi_j(\mathbf{z}_j) \leftarrow \min_{\phi_j} \sum_{i=1}^{n} \left[(y_i - \sum_{k \neq j} \phi_k(\mathbf{z}_{ik})) - \phi_j(\mathbf{z}_{ij}) \right]^2$$

until convergence.

The most extensively studied low dimensional expansion is the additive model

$$f(\mathbf{x}) = \sum_{j=1}^{p} \phi_j(x_j).$$

The popularity of the additive model is due to the local approximation methods, which work best for one-dimensional problems, and to the limited maximum number of elements that can enter the approximating relationship, namely the total number of covariates p.

Approximating multivariate regression functions by low dimensional expansions has some limitations. In practice the difficulties that local approximation methods generally have with variable spaces of dimension higher than two, limit the dimensionality of the expansion functions $\phi(\mathbf{z}_j)$ to two, at maximum. This implies that no more than two interacting variables can be present in the final approximation. When the 'true' regression function consists of more than two interacting terms, a misspecified model is the consequence. Performance and computational considerations constrain the number of expansion functions J that could potentially be entered to a small subset. This subset will depend on the true underlying function $g(\mathbf{x})$, and is generally unknown.

2.2.3. Adaptive Computation

In contrast with the low dimensional expansion methods, methods based on adaptive computation dynamically adjust their approximation strategy by taking into account the behaviour of the particular function to be modelled. Low dimensional expansions are in a sense nonadaptive, since their computing strategy is independent of the true function to be modelled; the subset of variables is, after selection, used in the expansion functions to construct the approximating function.

Adaptive algorithms can be subdivided into two groups, based on the strategies they use to reduce dimensionality. The general idea is to limit dimensionality without restricting the number of interacting variables beforehand. The first paradigm is known as *recursive partitioning*, the second one as *projection pursuit*.

2.2.3.1. Recursive Partitioning

The general idea behind recursive partitioning is to recursively split the entire covariate space $D \subseteq \Re^p$ up into several subregions, and to approximate the part of the true underlying function g that lies within a specific subregion by a function that depends on only a few of the total set of variables. Recursive partitioning *regression* [BFOS84] uses linear parametric or constant approximating functions within each subregion.

The recursive partitioning regression model approximates the underlying function by a function of the form

$$\text{if } \boldsymbol{x} \in R_m, \text{ then } f(\mathbf{x}) = \phi_m(\mathbf{x}),$$

where R_m are disjoint subregions partitioning the domain D. The functions ϕ_m usually have a simple *parametric* form, e.g., linear or a constant. The goal is to use the data to simultaneously estimate both a good set of subregions and the parameters associated with the separate functions in each subregion.

This procedure partitions the variable space by recursively splitting previous subregions. The starting region is the entire domain D. At each stage of the partitioning, each subregion is optimally split into two daughter subregions. A region R is split in the following way:

$$\begin{aligned} &\text{if } \mathbf{x} \in R, \text{ then} \\ &\quad \text{if } x_v \leq t, \text{ then } \mathbf{x} \in R_l \\ &\qquad\qquad \text{else } \mathbf{x} \in R_r, \end{aligned}$$

where v labels one of the covariates (*splitting variable*) and t is a value on that variable *splitting value*). The split is jointly optimised over v and t, using a goodness-of-fit criterion on the resulting approximation. The recursive subdivision stops as soon as some prespecified number of subregions has been generated. The subregions are then recombined in a reverse manner until an 'optimal' set is reached, based on some criterion (see [BFOS84] for a detailed description).

These recursive partitioning methods can be viewed as local averaging procedures, but unlike kernel and nearest neighbour procedures, the local regions are adaptively constructed based on the nature of the response variation. In many situations, this procedure results in improved performance.

Advantages of recursive partitioning regression are ease of interpretation, ease of computation, and ease of evaluation [Fri91]. The disadvantages are discontinuities at the boundaries of the subregions, difficulties in approximating certain types of simple functions (linear and additive in many variables) and in approximating functions with dominant interactions involving only a small fraction of the total number of variables. Additionally, the linear parts and the complex interactions cannot be discerned from the representation [Fri91].

The method called Multivariate Adaptive Regression Splines (MARS) [Fri91] is designed to overcome some of the limitations of general recursive partitioning regression. It can be seen as a generalisation of the latter procedure. In MARS the basis functions are splines instead of constants. The reader is directed to [Fri91] for an in depth discussion of MARS.

2.2.3.2. Projection pursuit

The general idea behind projection pursuit is to enable function approximation in high dimensional spaces with only moderate data supply, by projecting the data onto 'interesting' directions in order to discern a lower-dimensional pattern in the underlying function. Both projection pursuit regression (PPR) [FS81] and neural network learning (NNL) [Rip93c, HLMS93] are based on this principle.

PPR tries to approximate the underlying function $g(\mathbf{x})$ by a sum of ridge functions ϕ_m that are constant in a certain direction in variable space:

$$f(\mathbf{x}) = \sum_{m=1}^{M} \phi_m(\boldsymbol{\alpha}_m^T \mathbf{x}).$$

The approximation is constructed in an iterative manner. Residuals r_i are initialised by the y-values. The next term $\phi_k(\boldsymbol{\alpha}_k^T \mathbf{x}_i)$ in the model is determined as follows. For a given linear combination $\boldsymbol{\alpha}^T \mathbf{x}$, construct a smooth representation $\phi(\boldsymbol{\alpha}^T \mathbf{x})$ of the current residuals that are ordered in ascending value of $\boldsymbol{\alpha}^T \mathbf{x}$. This proceeds in a univariate setting, so general scatterplot smoothing techniques can be employed. At iteration k the linear combination $\boldsymbol{\alpha}_k$ is determined by maximising the criterion of fit $I(\boldsymbol{\alpha})$

$$I(\boldsymbol{\alpha}_k) = 1 - \sum_{i=1}^{n} [r_i - \phi_k(\boldsymbol{\alpha}_k^T \mathbf{x}_i)]^2 / \sum_{i=1}^{n} r_i^2,$$

which represents the fraction of so far unexplained variance that is explained by $\phi_k(\boldsymbol{\alpha}_k^T \mathbf{x}_i)$. The criterion of fit is maximised by some numerical optimisation method, which at each step needs to construct the smoother ϕ_k. When for iteration k the optimal linear combination has been found, the ridge function $\phi_k(\boldsymbol{\alpha}_k^T \mathbf{x})$ is added to the total sum of ridge functions, and new residuals r_i are calculated by

$$r_i = y_i - \sum_{j=1}^{k} \phi_j(\alpha_j^T \mathbf{x}_i).$$

This procedure terminates when the criterion of fit is smaller than a user-specified threshold –the last term is not included in the model.

In practice, in the early stages of the procedure the unexplained part of the variability of f can be quite large, and the smoothing is correspondingly unreliable. Therefore, when a new ridge function has been added, backfitting is used to reoptimise the earlier summands ϕ_j (and possibly also the α_j) in turn, keeping the other $k - 1$ contributions fixed.

PPR can be viewed as a low dimensional expansion method in which the (one-dimensional) arguments are not prespecified, but are dynamically constructed from the data set at hand. In this way many limitations of the other nonparametric regression techniques are overcome.

The sparsity limitation of kernel and nearest-neighbour techniques is not encountered, since all estimation (smoothing) is performed in a univariate setting. Unlike recursive partitioning, PPR does not split the data sample, thereby allowing, when necessary, more complex models. In addition, interactions of predictor variables are indirectly considered.

A disadvantages of PPR is that there exist some simple functions that require a large number of ridge functions for good approximation, e.g. [Hub85],

$$g(\mathbf{x}) = e^{x_1 x_2}.$$

Further, it is difficult to separate the linear from the interaction effects associated with the variable dependencies. Interpretation of the approximating function is difficult when the number of ridge functions is large. Constructing the approximation is time consuming [Fri91]. Additionally, the choice of the bandwidth of the smoother used to find ϕ_k is very delicate.

Another technique that in principle uses the same dimensionality-reduction strategy is the feed-forward neural network, which will be the subject of the subsequent chapters. A single layer feed-forward neural network with one linear output unit tries to approximate $g(\mathbf{x})$ by a composition of nonlinear signals

$$f(\mathbf{x}) = \sum_{m=1}^{M} w_m \phi(\boldsymbol{\alpha}_m^T \mathbf{x}_i)$$

where ϕ is a univariate nonlinear function, which transfers the projected input vector. Unlike the ϕ-functions in PPR, these functions are of a fixed form, usually sigmoid, and are selected independently from the data before actual network training starts. The subsequent chapters are dedicated to neural networks, therefore we will refrain from an elaborate discussion of this technique at this point.

There is a clear conceptual resemblance between neural network regression and PPR [HLMS93]. It is possible to implement PPR by a neural network architecture [VD94]. PPR and neural network regression differ in the way parameters are estimated and in the type of expansion functions used. All parameters (weights) in a neural network are simultaneously adapted during learning, whereas in PPR the parameters α_m and the smoothing functions ϕ_m are iteratively adapted. PPR builds an approximating surface out of flexible (smoothing) functions, whereas neural network regression builds this surface out of prespecified fixed "squashing" functions.

2.3. The Bias/Variance Dilemma

In the previous section several techniques were introduced that approximate $g(\mathbf{x}) \equiv E[y|\mathbf{x}]$ in a flexible way. These techniques intensively use the sample data set $D = \{(\mathbf{x}_i, y_i)\}_1^n$ in constructing the approximating function $f(\mathbf{x})$. This section deals with the *bias/variance dilemma*,

a major statistical problem that is inherent to flexible regression modelling. The essence of the dilemma lies is that the approximation error can be decomposed into two components, known as the bias and the variance. In the practice of data modelling, approximating functions that have low bias generally have high variance, whereas approximating functions that have low variance generally have high bias. The consistency feature of the flexible regression techniques implies small bias; but when faced with limited data, high variance is often the consequence. Constraining the level of flexibility reduces the variance, but also implies a larger bias. The foregoing illustrates the dilemma: finding a compromise between bias and variance.

To assess the effectiveness of $f(\mathbf{x})$ as predictor of y given $\mathbf{x}$, we use the conditional mean squared error criterion $E[(Y - f(\mathbf{x}))^2|\mathbf{x}]$, which can be rewritten as

$$E[(Y - f(\mathbf{x}))^2|\mathbf{x}] = E[(Y - g(\mathbf{x}))^2|\mathbf{x}] + (f(\mathbf{x}) - g(\mathbf{x}))^2. \tag{2.2}$$

Equation (2.2) reveals that one part of the expected error (given $\mathbf{x}$) is completely determined by the conditional variance of y given $\mathbf{x}$, and the other is determined by the deviation of the predictor $f(\mathbf{x})$ from $g(\mathbf{x})$. It shows that minimum expected squared error is achieved if $f(\mathbf{x})$ approximates $g(\mathbf{x})$ as close as possible.

We use the mean squared error to assess how effectively $f(\mathbf{x})$ approximates $g(\mathbf{x})$. Since in general $f(\mathbf{x})$ is constructed on a finite data set $\mathcal{D}_n$, this $\mathcal{D}_n$ is explicitly incorporated in the description of the approximating function. The mean squared error is defined as the average value of $(f(\mathbf{x};\mathcal{D}_n) - g(\mathbf{x}))^2$ when the data set $\mathcal{D}_n$ is repeatedly constructed by independent drawings from a joint probability distribution $p_{\mathbf{x}y}$. For any $\mathbf{x}$, we obtain ([GBD92])

$$\begin{aligned} E_{\mathcal{D}_n}\left[(f(\mathbf{x};\mathcal{D}_n) - g(\mathbf{x}))^2\right] &= (E_{\mathcal{D}_n}[f(\mathbf{x};\mathcal{D}_n)] - g(\mathbf{x}))^2 \\ &\quad + E_{\mathcal{D}_n}\left[(f(\mathbf{x};\mathcal{D}_n) - E_{\mathcal{D}_n}[f(\mathbf{x};\mathcal{D}_n)])^2\right], \end{aligned} \tag{2.3}$$

where $E_{\mathcal{D}_n}$ denotes the expectation with respect to the probability distribution of $\mathcal{D}_n$. The first part of the decomposition is the "bias" part; the second part is the "variance" part. The bias part shows the deviation of the average predictor value from $g(\mathbf{x})$, also called the accuracy of the predictor. The variance part, or precision, shows the average squared distance of the predictor from its own average. For example, a highly flexible method that simply interpolates the observations in each $\mathcal{D}_n$ will be asymptotically unbiased, since each time it represents the $(\mathbf{x}_i, y_i)$ patterns of subset $\mathcal{D}_n$ drawn from the *population* distribution $p_{\mathbf{x}y}$; averaging over all possible subsets asymptotically approaches $E[Y|\mathbf{x}]$. The variance, on the other hand, will be high. In case of exact interpolation, the variance equals the conditional variance of Y.

It is clear that the bias and the variance of an estimator typically are concepts that have a meaning only in the repeated sampling approach to statistics. Although unbiasedness plays an important role in statistical inference, it may be better to accept some bias when it can be traded against lower variance. This is the reason why linear models sometimes perform reasonably

well, even when it is suspected that the true underlying function $g(\mathbf{x})$ is nonlinear but the type of nonlinearity is unknown.

2.4. Cross-validatory Choice of Flexibility Parameters

The bias/variance dilemma implies that reducing the flexibility of a method may be necessary to obtain useful predictions. All model free regression methods have parameters that influence the flexibility of the resulting fit. These flexibility parameters are a multivariate variant of the *smoothing parameters* used in univariate regression modelling or scatterplot smoothing [HT90, Hae90]. In nearest neighbour regression, for example, a neighbourhood size of one makes the fit very flexible, whereas a neighbourhood size equal to the total sample size makes the fit rigid, namely the average y-value of the complete data set.

Let θ denote the 'flexibility' parameters of a flexible regression function f. The θ-parameters have to be set using the data set $D = \{(\mathbf{x}_i, y_i)\}_1^n$ at hand. Good parameter values are obtained by minimising some global error measure, such as *average mean squared error*

$$MSE(\theta) = 1/n \sum_{i=1}^{n} E_{\mathcal{D}_n} \left[(f_\theta(\mathbf{x}_i; \mathcal{D}_n) - g(\mathbf{x}_i))^2 \right]. \tag{2.4}$$

To be explicit about the dependence of f on the parameters θ and the data sample at hand, we write $f_\theta(\mathbf{x}_i; \mathcal{D}_n)$ instead of $f(\mathbf{x}_i)$. $E_{\mathcal{D}_n}$ indicates the statistical expectation taken over all subsets of size n from the total population. In general, however, the *true* regression function $g(\mathbf{x})$ is unknown; calculating MSE is therefore impossible. Another measure that differs from MSE by only a constant function $\sigma^2 = \mathrm{Var}(\epsilon)$ is the *average predictive squared error*

$$PSE(\theta) = 1/n \sum_{i=1}^{n} E_{\mathcal{D}_n} \left[(f_\theta(\mathbf{x}_i; \mathcal{D}_n) - y_i^*)^2 \right], \tag{2.5}$$

where y_i^* is a new observation at $\mathbf{x}_i$, that is, $y_i^* = g(\mathbf{x}_i) + \epsilon_i^*$. It can be easily shown that $PSE = MSE + \sigma^2$.

Notice that in these summary measures we are conditioning on the observed values of the data set D at hand. An alternative to (2.4), which is theoretically preferable, minimises the expected mean squared error averaged over the true distribution of X

$$MSE(\theta) = \int_{\mathbf{x}} E_{\mathcal{D}_n} \left[(f_\theta(\mathbf{x}; \mathcal{D}_n) - g(\mathbf{x}))^2 \right] p_{\mathbf{x}} d\mathbf{x}. \tag{2.6}$$

This measure is computationally very demanding, if not impossible to compute. So, for the remainder we use PSE as defined in (2.5). Good parameter values minimise PSE.

Since we usually do not have repeated measurements at the particular $\mathbf{x}$-values, formula (2.5) can not be expressed analytically. Therefore, we have to approximate the prediction error

PSE. The standard method that uses a hold-out set is not advisable when the data set is small. Hence, only part of the data is used for training, whereas one would like to use as many data as possible to reduce estimation errors. The observations held out from the training set are used to estimate the prediction error. If the hold-out set is too small, a precise (low variance) estimate of the prediction error can not be obtained.

Cross-validation is an alternative method for estimating the prediction error of a regression function [Sto74, MU94, HT90, Koh95]. It makes no assumptions about the statistics of the data. Standard cross-validation works by leaving out points $(\mathbf{x}_i, y_i)$ one at a time, and constructing a regression on the remaining $n-1$ points. This is an attempt to mimic the repeated use of one data set for training and one other data set for prediction. The *average cross-validation squared error*

$$CV(\theta) = 1/n \sum_{i=1}^{n} \left(f_\theta(\mathbf{x}_i; D^{-i}) - y_i \right)^2 \tag{2.7}$$

is then constructed, where $f(\mathbf{x}; D^{-i})$ indicates that f is fitted on the data set D without observation $(\mathbf{x}_i, y_i)$. This form of cross-validation is known as *leave-one-out*. It has two disadvantages. First, $CV(\theta)$ can be expensive to compute. Second, leave-one-out –although nearly unbiased– shows high variance [Efr83, WK91], which makes it difficult to correctly choose parameter values.

The leaving-one-out estimator is a special case of the general class of k-fold cross-validation error rate estimators. In k-fold cross-validation, the data are randomly divided into mutually exclusive test partitions of approximately equal size. The patterns not present in each test partition are independently used for training, and the resulting regression function is tested on the corresponding test partition. The cross-validated error rate is the average error rate over all k partitions. Kohavi [Koh95] and Zhang [Zha93] suggest to use ten or five folds.

The main reason for estimating the prediction errors was to use them in selecting good values for the flexibility parameters. What we have discussed above is known as the cross-validatory choice of parameters. Cross-validation, of course, can also be used for the assessment of statistical prediction. However, one should guard against the interference of both. As an example, a researcher constructs a prediction model for a particular phenomenon, on which he has n observations, by some flexible regression technique. He obtains good values θ^* for the flexibility parameters by minimising the cross-validation error $CV(\theta)$. In an article he describes the final solution, reports the $CV(\theta^*)$ as the expected prediction performance, and compares this performance to the performances achieved by alternative models. The flaw in this approach is that the expected prediction accuracy is optimistically biased, since the flexibility parameters were obtained by minimising just this prediction measure.

There are at least two solutions to this problem that give an honest estimation of expected performance. First, estimate the expected prediction performance on a hold-out set *not* used in

the procedure of cross-validatory parameter choice. With limited data this option is unattractive, since a significantly large part of the data is not used for model building. The hold-out set must have considerable size; otherwise the variance in the estimate of the prediction performance will be too large to be useful. A second approach is the 'two-deep' cross-validation procedure described in [Sto74], which proceeds as follows. The total data sample is randomly divided into k subsets. Each subset is used once for the calculation of the prediction error of the model constructed on the remaining subsets. The construction of the model on these remaining subsets is again guided by cross-validation, i.e., the subsample is randomly divided into k' subsets, and the usual procedure of model selection is applied. This approach requires $k \times k'$ models to be constructed. For many model free methods, the two-deep cross-validation procedure takes too much computing time.

2.5. Conclusions

The recent increase in available computer power has made flexible regression a feasible alternative to linear modelling, especially when there is uncertainty about the functional form specification. We have introduced the ideas behind some well known flexible regression methodologies. Feed-forward neural networks, which will be investigated profoundly in the remaining chapters, were introduced briefly. This chapter primarily showed that several strategies can be followed to find a flexible approximation to a particular underlying relationship, and that a neural network is one of them. The most promising strategies in high-dimensional problems with relatively few observations are based on adaptive computation. Recursive partitioning and projections pursuit are two strategies to reduce the dimension of the initial regression problem. Neural network regression and projection pursuit regression are examples of the latter strategy.

All members of the general class of flexible regression methodologies –neural networks included– suffer from the principal difficulties caused by the bias/variance dilemma. This means that the price one has to pay for a decrease in the bias of an estimator of the true $g(\mathbf{x})$, usually is an increase in the variance of the estimator. While being on average (over many repetitions) closer to the true underlying function, the resulting approximating functions are more spread apart, which increases the risk of making bad predictions, when a regression model is fitted to a particular data sample. This phenomenon is a very important determinant of the practical success of flexible regression methods, including neural networks.

Flexible regression techniques have at least one parameter that determines the degree of flexibility (smoothness) of the resulting model. We discussed a general resampling method, known as cross-validation, that enables the selection of a good value for the flexibility parameter(s). In the remainder we will adopt the cross-validation approach to select parameters in a neural network.

Chapter 3

Theoretical Aspects of Neural Networks

3.1. Introduction

This chapter focusses on neural networks, which are particular members from the class of flexible regression methods, introduced in the previous chapter. The theoretical aspects of neural network learning are discussed from a statistical perspective.

Neural networks are a class of input-output models, also called information processing systems, originated from cognitive science. In this scientific area, researchers try to understand how the human brain (or human intelligence) works –how it stores information, how it retrieves information, and how it learns. Humans provide the best example of intelligent systems, so attempting to build intelligent machines that 'act like humans' is not a futile activity.

The computer provided cognitive scientists with a means to actually build models of the brain and to use them in the study of the brain's main functioning. Artificial neural networks are used to simulate learning strategies of the mind when provided with learning examples. The neural network's topology is an abstract 'translation' of the elements of the human brain. The brain consists of about 10^{11} neurons (miniature communication devices), which are highly interconnected by links called synapses, through which signals are submitted and received. Since many authors, among others [HKP91, ARe88], have already elaborated on the mapping of elements and processes of the biological brain onto the elements of an artificial neural network, this discussion will not be repeated here.

The architecture, or topology, of artificial neural networks has evolved from simple perceptrons to multilayer (recurrent) neural networks and to more complicated structures. In [ARe88] a good picture of the evolution of neural networks is sketched in a collection of papers that were of great importance for the neural network field. A good discussion of the various types of artificial neural networks and learning strategies is given in [HKP91]. We focus on feed-forward neural networks, which are most popular in applications.

Although inspired by certain aspects of information processing in the brain, the neural network models and their related learning paradigms are still far away from a realistic description of how the human brain works. Nevertheless, as a data analysis tool they have proven qualities in different applications, especially in pattern recognition tasks. In this guise artificial neural networks are more and more often applied to economic and financial modelling problems. The literature on neural networks, however, remains confused as to whether artificial neural networks are supposed to be realistic biological models or practical machines. For data analysis, biological plausibility is irrelevant. Therefore, in what follows we will refrain from any biological plausibility, and imply the adjective "artificial"; we will speak of neural networks (NN) for short.

The outline of this chapter is as follows. In section 2 feed-forward neural networks are represented in graphical and in mathematical ways. In section 3 the statistical aspects of neural network learning are discussed. Section 4 addresses the generalisation issue. Section 5 discusses how to compare the predictive performance of neural network models with the predictive performance of alternative methods in a statistically sound manner. Section 6 concludes the chapter.

3.2. Graphical and Mathematical Representation of NN

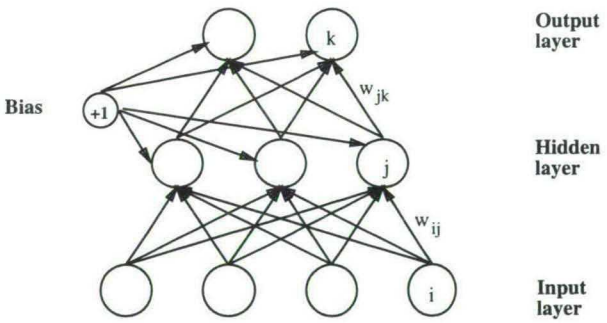

Figure 3.1: A generic feed-forward neural network with a single hidden layer; the bias neuron has been removed from the input layer.

A neural network model is a particular type of input-output model. Given an input vector $\mathbf{x} = (x_1, \ldots, x_p)'$ the network produces an output vector $\hat{\mathbf{y}} = (\hat{y}_1, \ldots, \hat{y}_q)'$. In statistics it is common practice use a hat to denote estimated variables, which NN outputs in fact are. We conform to this notation.

A widely studied network is the feed-forward neural network; an example is depicted in Figure 3.1. In graphical form, feed-forward neural networks consist of directed graphs without cycles. Each node represents a "unit", also called artificial neuron, which is the building brick of the artificial neural network. The functionality of each unit is as follows. Each non-input unit j sums its incoming signals and adds a constant term (the bias or intercept in statistical terminology) to form the total incoming signal and applies a function ϕ to this total incoming signal to construct the output of the unit. The links have weights w_{ij} which multiply the signal travelling through them by that factor. Figure 3.2 shows the functionality of an artificial neuron. The function ϕ is called the transfer, activation, or squashing function; ϕ is usually taken to be logistic (with $\phi(z) = \frac{\exp(z)}{1+\exp(z)}$), or threshold (with $\phi(x) = I(x > 0)$).

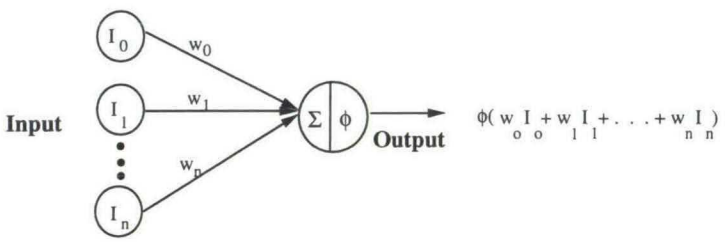

Figure 3.2: Graphical representation of a neuron

The input units only distribute the input vector, so their ϕ is the identity function. Implementing a bias term in the network is done by the addition of an "extra" input unit that always has the value of 1, and that is connected to each unit from the hidden and output layer. Now each bias term becomes an ordinary weight, and we do not need to distinguish between bias terms and ordinary weights.

In mathematical notation a feed-forward neural network, as depicted in Figure 3.1, is expressed by

$$\hat{y}_k = \phi_O(\sum_{j \to k} w_{jk} \phi_H(\sum_{i \to j} w_{ij} x_i)), \tag{3.1}$$

where we used $\hat{y}_k$ to denote the value of the k'th output unit when input x is fed into the network, and $\sum_{i \to j}$ stands for the sum over neurons i connected to j. Usually identical squashing functions are used within the same layer; ϕ_O denotes the squashing function of the output units, and ϕ_H denotes the squashing function of the hidden units. The indexing presumes that neurons are numbered sequentially in the order: input units, hidden units, and output unit(s).

The mathematical formulation of the feed-forward NN model shows great resemblance with projection pursuit models (see Chapter 2). The main difference is the squashing functions, which are fixed for a NN and free for projection pursuit regression. When the number of hidden nodes is fixed in both PPR and NN, then PPR can approximate a larger class of functions ([HLMS93]).

The parameterised representation of a neural network allows for easy and fast calculation of predictions.

It may be preferable to have the neural network include direct 'skip-layer' connections from the input layer to the output layer to explicitly incorporate the basic linear model, which leads to

$$\hat{y}_k = \phi_O(\sum_{i \to k} w_{ik} x_i + \sum_{j \to k} w_{jk} \phi_H(\sum_{i \to j} w_{ij} x_i)). \tag{3.2}$$

In the remainder the expression $f(\mathbf{x}, \mathbf{w})$ is used as short-hand for the network output function, where $\mathbf{x}$ represents the input vector and $\mathbf{w}$ the vector of all the weights. This notation is convenient since it depends only on inputs and weights, given a fixed network architecture. From now on the dimension q of the output vector $\hat{\mathbf{y}}$ is assumed to be one, that is, our neural network consists of a single output unit. This assumption avoids unnecessary complex expressions in the remaining, not loosing generality. Further, the squashing function of the output unit ϕ_O is assumed to be linear, which enlarges the resemblance with the alternative regression forms. In the remainder we let ϕ (without subscript) denote the squashing function of the hidden units.

3.3. Neural Network Learning

Equations (3.1) and (3.2) represent quite general classes of functions. A number of authors (e.g., [Cyb89]) have shown that feed-forward neural networks with a single hidden layer and nonlinear squashing functions ϕ (e.g., sigmoid) for the hidden units can approximate any *continuous* function g uniformly on compact sets, by increasing the size of the hidden layer; the squashing functions of the output unit(s) may be linear.

Given a network with a sufficient number of hidden units, the role of learning is to find suitable values for the network weights $\mathbf{w}$ to approximate a function g of $\mathbf{x}$ by $f(\mathbf{x}, \mathbf{w})$. The estimation of the weights has been the main reason for stagnation in neural network research for many years. In the next subsection neural network learning is discussed from a statistical perspective.

3.3.1. A Statistical Approach

The premise of this section is that learning procedures used to train neural networks, are inherently statistical techniques. Given a fixed neural network architecture, the output function $f(\mathbf{x}, \mathbf{w})$ can be viewed as a parametrised nonlinear form that has to be fitted to the data, as in nonlinear regression. This observation suggests that we can apply the principles of nonlinear regression analysis, which is a well researched area in statistics; see, for example, [DD88].

The theory that is presented here is based on the work done by White [Whi89b], who discusses neural network learning in a statistical framework. White argues that learning in neural networks when optimising some performance measurement is implicitly directed to the discovery of certain aspects of the conditional probability law $p(y|\mathbf{x})$. We are interested in the relationship between X and Y, because X is used to predict Y. In such a case, network performance can be measured using a performance function π, also called a loss or error function. Given a target value y and the network output $\hat{y}$, the performance function gives a numerical value $\pi[y, \hat{y}]$ that indicates how well the network performs (on the training data). Usually a larger π-value means that the network performance becomes worse. The most frequently used performance function is

$$\pi[y, \hat{y}] = (y - \hat{y})^2,$$

although many other choices are possible, e.g., $\pi[y, \hat{y}] = |y - \hat{y}|^k/k$ $(k > 0)$. Berger [Ber85] provides the theory on loss functions in statistical decision theory. Once the neural network architecture f, weights $\mathbf{w}$, target values y, and network inputs $\mathbf{x}$ are specified, the network performance is measured by $\pi[y, f(\mathbf{x}, \mathbf{w})]$.

It is usually required that the neural network performs well over a whole range of situations, that is, for new $\mathbf{x}$ and y values. In statistical terms, we want the network to perform well on average. Average performance is given mathematically by

$$\rho(\mathbf{w}) = \int \pi[y, f(\mathbf{x}, \mathbf{w})] p(\mathbf{x}, y) d\mathbf{x} dy \tag{3.3}$$

$$\equiv E[\pi[Y, f(X, \mathbf{w})]], \tag{3.4}$$

where the network architecture f and $\mathbf{w}$ are fixed. We call ρ the expected performance function, which corresponds to the risk function in statistical decision theory. Note that given a specific network architecture, ρ depends only on the weights $\mathbf{w}$ and not on the $\mathbf{x}$ and y, which have been averaged out. Each particular weight vector $\mathbf{w}$ will lead to a different expected performance $\rho(\mathbf{w})$. The goal of learning is to find that weight vector $\mathbf{w}^*$ that minimises $\rho(\mathbf{w})$. We will refer to $\mathbf{w}^*$ as the "optimal weights" vector, which is not necessarily unique. It should be noted that instead of requiring optimal average performance, we could have selected any other global performance measure, for instance, median performance. We shall continue to use the average performance interpretation, since it is the standard.

Note that the joint probability law $p(\mathbf{x})$, plays an important role in determining the optimal weight vector $\mathbf{w}^*$. These weights give small errors for X values that are likely to occur (according to $p(\mathbf{x})$) at the cost of larger errors (on average) for X values that are not likely to occur. It is evident that the optimal weights $\mathbf{w}^*$ perform optimally in practice only when π and $p(\mathbf{x})$ are selected in such a way that they reflect accurately the conditions met in practice.

If the joint probability law $p(\mathbf{x}, y)$ were known, we could directly solve (3.4) for $\mathbf{w}^*$. It is the lack of knowledge on $p(\mathbf{x}, y)$ that makes learning necessary. In economics we gather empirical

knowledge on $p(\mathbf{x}, y)$ by making repeated measurements on X and Y. In practice we have a finite sample from which we gain information. Based on a sample $D_n = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, we calculate the sample analog of $p(\mathbf{x}, y)$, denoted by $p_n(\mathbf{x}, y)$, as follows:

$$p_n(C) \equiv 1/n \times (\text{number of times } (\mathbf{x}_t, y_t) \text{ belongs to } C)$$

where C is any (Borel measurable) subset from $\Re^{p+1}$. When n is large, the law of large numbers[1] ensures that $p_n(C)$ is a good approximation to $p(C)$. Using this approximation, we calculate the approximation ρ_n to ρ by

$$\rho_n(\mathbf{w}) \equiv \int_{(\mathbf{x},y)\in D_n} \pi[y, f(\mathbf{x}, \mathbf{w})] p_n(\mathbf{x}, y) d\mathbf{x} dy \tag{3.5}$$

$$= 1/n \sum_{i=1}^{n} \pi[y_i, f(\mathbf{x}_i, \mathbf{w})], \tag{3.6}$$

which is the average performance of the neural network over the training sample D_n.

The value corresponding to (3.6) is easily calculated, so we can determine the weight vector $\mathbf{w}_n$ by solving

$$\min_{\mathbf{w}} \rho_n(\mathbf{w}).$$

The vector $\mathbf{w}_n$ actually is a realisation of a random variable; hence, each time a new training set D_n is drawn, the vector $\mathbf{w}_n$ will change. This prevents us from making more than probabilistic statements about the true optimal weight vector $\mathbf{w}^*$.

Using (3.6) with random counter parts of the variables involved, we define $\hat{\mathbf{w}}_n$ as the random variable that solves the problem

$$\min_{\mathbf{w}} \hat{\rho}_n(\mathbf{w}) = 1/n \sum_{i=1}^{n} \pi[Y_i, f(X_i, \mathbf{w})],$$

where $\hat{\rho}_n(\mathbf{w})$ is a stochastic variable due to the randomness of X_i and Y_i.

In the special case of squared error loss ($\pi[y, \hat{y}] = [y - \hat{y}]^2/2$) we get

$$\min_{\mathbf{w}} 1/n \sum_{i=1}^{n} [Y_i - f(X_i, \mathbf{w})]^2/2.$$

This is precisely the problem of nonlinear least-squares regression, so the resulting $\hat{\mathbf{w}}_n$ is a nonlinear least-squares estimator. Nonlinear regression has been extensively analysed in econometrics and statistics, e.g., [DD88, JG85]. The neural network community has developed its 'own' solution to this minimisation problem, known as *error back-propagation*, which is discussed in the next section.

[1] which requires that $\mathbf{x}_t$ and y_t are asymptotically independent

From this point the standard statistical approach to nonlinear regression can be followed. First, derive the limiting distribution for $\hat{\mathbf{w}}_n$, which is approximately multivariate normal (see [Whi89b] for details). Then, use this limiting distribution to specify approximate confidence intervals for the weights, or to test specific hypotheses, such as testing for irrelevant hidden units or irrelevant inputs [KW92]. Also prediction intervals can be constructed, using the limiting distribution for the weights. According to White [Whi89b], the statistical inference approach to neural networks is to be preferred, although in the neural network community the significance of this approach has not (yet) been widely appreciated or exploited. The reason seems to be the gap between theory and practice. In practice, it is not known how large n must be to ensure a good approximation, and the answer is highly context dependent. In general, the more weights a neural network contains, the higher n must be to obtain a given degree of approximation. In applications the number of weights often exceeds the number of training cases available. The same issue is also raised by Ripley [Rip93a]: "In non-linear regression (Bates and Watts, 1988) we would attempt to quantify the uncertainty in the parameters and in the predictions,... Until recently neural networks had not been considered in the same light. One problem is that they tend to have very large numbers of parameters relative to the number of training cases, and the parameters are not meaningful, so error statements for the weights are less useful than for parameters in mechanistically-specified non-linear regressions". While this is true, it may be interesting to look at certain functions of the weights, for example, $\frac{\partial f}{\partial x_i}|_{\overline{\mathbf{x}}}$ or $E[\frac{\partial f}{\partial x_i}]$.

3.3.2. Minimisation

The foregoing subsection provided the statistical rationale for the determination of the weight vector $\mathbf{w}$ by minimising

$$\sum_{i=1}^{n}[y_i - f(\mathbf{x}_i, \mathbf{w})]^2. \tag{3.7}$$

Error back-propagation, the best-known learning method in the neural network community, permits weights to be learned from experience in a process resembling trial and error ([RHW86]). Experience is based on empirical observations on the phenomenon of interest. Since error back-propagation is extensively described in almost all textbooks on neural networks, we will be very short on it. Back-propagation simply is the application of the gradient descent technique to the minimisation of (3.7). According to back-propagation, we start with a set of random weights $\mathbf{w}_0$ and then update them by the formula

$$\mathbf{w}_l = \mathbf{w}_{l-1} + \eta \sum_{i=1}^{n} \nabla f(\mathbf{x}_i, \mathbf{w}_{l-1})(y_i - f(\mathbf{x}_i, \mathbf{w}_{l-1})),\ l = 1,\ 2, \ldots \tag{3.8}$$

where η is a learning rate and ∇f is the gradient (the vector containing the partial derivatives) of f with respect to the weights $\mathbf{w}$ ([RHW86]). In the neural network community (3.8) is known

as *batch learning*; the weights are updated after each complete presentation of the training set (of size n). Opposed to batch learning is *incremental* or *on-line* updating of the weights, that is, all weights are updated after the presentation of a single input pattern $\mathbf{x}_i$ ($i = 1, 2, \ldots, n$). When the latter strategy is used, a *momentum* term is often added to the update rule for the weights. This can be seen as applying exponential smoothing to (3.8), and leads to

$$\Delta \mathbf{w}_l = -(1-\alpha)\eta \sum_{i=1}^{n} \nabla f(\mathbf{x}_i, \mathbf{w}_{l-1})(y_i - f(\mathbf{x}_i, \mathbf{w}_{l-1})) + \alpha(\Delta \mathbf{w}_{l-1}).$$

So, a momentum term multiplies the previous weights' change with a factor α that lies between zero and one. A large value of the momentum term, say, $\alpha = 0.9$, makes the next weights update resemble the current weights update.

The calculation of $\nabla f(\mathbf{x}, \mathbf{w})(y - f(\mathbf{x}, \mathbf{w}))$ is performed by making only local computations on the network itself. So, the specific structure of a neural network is used for the calculation of the gradient of the error function (3.7). We will not elaborate on this topic here; for a good description of feed-forward neural networks and its specific learning algorithms we refer to [HKP91].

The problem of minimising (3.7) is characterised by the presence of locally optimal weights. Backpropagation as well as its competitors from statistics can become 'trapped' in these local minima. In the neural network literature it is often presumed ([RHW86]) that the chance one ends up in a local minimum is low in practice. This premise, however, does not seem to be realistic. The next chapter exemplifies this. The *multi-start algorithm* is a simple heuristic that helps in finding 'good' local minima by making multiple restarts –each time with different random starting weights.

In case one is interested in the *global* minimum, one should be prepared to pay a high computing cost. Hence, global optimisation algorithms, such as simulated annealing [AK89, GFR94] or genetic algorithms [Gol89], in principle search the whole error space in a 'smart' way –inspired (again) by biological processes. In general these procedures are too computationally intensive to be practically useful for the purpose of weight estimation.

3.4. Generalisation

For small samples, neural networks just like all other flexible regression methods suffer from the *bias/variance dilemma* [GBD92], which we discussed in the previous chapter. When having a finite (medium or small sized) set of n observations, a neural network with a fixed architecture will use all its resources to make the fit to the data as good as possible. The danger, however, is that instead of the 'true' relationship between X and Y, a function is fitted through the n observations that approximates y as well as possible, including the disturbance term. If this

is the case, the neural network is said to "overfit" the data. Presenting n' new observations (generated by the same underlying system) to the trained network will result in a bad fit with high probability. When the network is retrained on these n' fresh observations, the network solution probably will differ considerably from the previous one. This is what high *variance* means in practice. When this retraining is repeated many times, the *average* network solution will be close to the true relationship: the network has a small *bias*.

In practice one is interested in networks that give good results for new data, that is, which *generalise* (or predict) well. One attempt to accomplish this goal is to reduce the variance by introducing a (small) bias. Smaller variance may be realised by preventing the weights from growing too large such that all individual observations cannot be approximated exactly. In back-propagation learning, this can be obtained by stopping the training before convergence has taken place. An independent test set is used to monitor the prediction error (see [FHZ93]) and to indicate when training must be stopped. Among statisticians this method receives little sympathy [Sar95]; the main reason is the subjectivity involved, and the dependence on the starting weights and on the particular test set chosen.

Another method that is used for the purpose of parameter restriction is *weight decay*. In statistics this method is better known as ridge regression; it is used in case of collinearity or near-collinearity of the independent variables. The idea behind weight decay is that instead of minimising (3.7), the following, adapted error function is minimised:

$$\sum_{i=1}^{n}[y_i - f(\mathbf{x}_i, \mathbf{w})]^2 + \lambda \sum_{ij} w_{ij}^2, \tag{3.9}$$

where the additional term penalises large weights (two small weights are preferred above one large weight). In Bayesian terminology, the weight decay term implements a prior distribution on the weights. Bayesian statistics illuminates the interpretation of the weight decay term. In Appendix A we elaborate on prediction and on the Bayesian perspective on neural network learning with a weight decay term. A practical disadvantage of weight decay could be the introduction of yet another parameter that has to be chosen *a priori*, namely λ. There will be more on this in Chapter 4.

3.5. Network Performance Analysis

When a final neural network has been constructed, the next step is to evaluate the performance of it. We adopt prediction quality as the main criterion, but other criteria, for example, interpretability, can be selected as well. Prediction performance, however, is less subjective than most other criteria.

The mutual discovery of the statistical and artificial intelligence communities (see, e.g., [Han93, CO94]) has resulted in many studies which compare the performance of statistical and

machine learning methods on empirical data sets; examples are the StatLog project ([MST94]) and the Santa Fe Time Series Competition ([WG94]), as well as numerous journal articles ([KWR93, RABCK93, WHR90, TAF91, TK92, FG93]).

We have observed that there is no consensus in the research community on how such a comparative study be performed in a methodologically sound way.

The ranking of k preselected methods is usually performed by training (estimating in statistical terminology) the methods on a single data set, and estimating their respective mean prediction errors (MPE) from a hold-out sample. The methods are subsequently ranked according to their estimated MPEs. Some studies use –in our view appropriately– statistical significance testing in order to make this ranking. However, the effect that comparing more than two methods has on the probability of generating a "false alarm" (claiming that one method is better than another, when in fact it is not) is to our knowledge ignored in this literature.

The statistical analysis of comparative studies -method ranking in particular- is addressed in the next two subsections, which are largely based on the study [FV95] by Feelders and Verkooijen. We address *methodological* issues of studies in which the performances of several regression methods are compared on empirical data sets. We first introduce some statistical terminology and concepts that are necessary to arrive at a useful multiple comparison procedure.

3.5.1. Pairwise Tests

The ranking of methods by simply ordering them by their estimated prediction errors should be extended by statistical significance testing. Appropriate tests are those for the difference between means (regression) and proportions (classification). The standard t-test for testing the difference between two sample means $\overline{Y}_1$ and $\overline{Y}_2$, which assumes *independent* normally distributed populations, leads to the following confidence interval for the difference

$$\theta_1 - \theta_2 \in [(\overline{Y}_1 - \overline{Y}_2) \pm t_{(\alpha/2,\nu)}\,\hat{\sigma}_{\text{diff}}], \tag{3.10}$$

where ν denotes the degrees of freedom, and $\hat{\sigma}_{\text{diff}}$ equals $\sqrt{\hat{\sigma}^2_{\overline{Y}_1} + \hat{\sigma}^2_{\overline{Y}_2}}$. In the standard comparative experiment, however, the MPEs are all estimated from the *same* test sample, which makes them highly correlated. Therefore, a *paired sample* t-test should be used instead. The dependence within the pairs only changes the standard error of the difference $\hat{\sigma}_{\text{diff}}$, which now becomes

$$\hat{\sigma}_{\text{diff}} = \sqrt{\hat{\sigma}^2_{\overline{Y}_1} + \hat{\sigma}^2_{\overline{Y}_2} - 2\,\text{cov}(\overline{Y}_1, \overline{Y}_2)}. \tag{3.11}$$

When the variables are positively correlated, the covariance has a positive value and thus the variance and standard error of a difference between means will be *smaller* for matched than for unmatched samples. Consequently, the confidence intervals become tighter (given the same α

value), which results in more powerful tests. In conclusion, neglecting the dependence between the samples generally results in too conservative tests.

The paired t-test becomes simpler when defining $D := Y_1 - Y_2$ and testing $H_0 : \theta = 0$ by

$$\theta \in [\overline{D} \pm t_{(\alpha/2,\nu)}\, \hat{\sigma}_{\overline{D}}]. \tag{3.12}$$

The main assumption behind the paired t-test for the difference between two means, is that the underlying population is normally distributed. When this assumption is not met, but the sample size n is large, then the central limit theorem justifies the application of the t-test. When, however, the sample size is medium or small, (3.12) may lead to wrong conclusions. In this case the following (bootstrap) resampling based t-test proposed by Westfall and Young [WY93, Algorithm 2.3] is recommended:

0. Calculate the statistic of interest $t = (\overline{d} - 0)/(s/\sqrt{n})$, where s is the sample deviation $s^2 = 1/n \sum_{i=1}^{n} (d_i - \overline{d})^2$.
1. Initialise the counting variable *count*.
2. Generate resample data $d_1^*, \ldots, d_n^*$ *with replacement* from the original data $d_1, \ldots, d_n$.
3. If
$$\frac{\overline{d}^* - \overline{d}}{s^*/\sqrt{n}} \geq t, \tag{3.13}$$
then $count \leftarrow count + 1$.
4. Repeat steps 2-3 N times. The estimated p-value is $p = count/N$.

3.5.2. Multiplicity Effect

Often the estimated MPEs of more than two, say k, methods are compared. The first idea that comes to mind is to test each possible difference by a paired t-test with a probability of Type I error of size α. The problem is that the probability of making at least one Type I error over the whole family of t-tests (one test per pair of methods being compared) exceeds α by an amount that increases with k (the number of tests made). For k *statistically independent* tests, the probability of making at least one Type I error, better known as the *familywise error* rate (FWE), is $1 - (1 - \alpha)^k$. When k is large, say 20, this can be a large probability; for $\alpha = 0.05$, there is a probability of 0.64 on one or more Type I errors. This means that the probability on *incorrectly* claiming the significance of at least one difference equals 0.64. Such an incorrect claim is often called a "false alarm". When the tests are statistically dependent on each other (such as is the case in pairwise difference tests) then the FWE becomes even larger. Thus, when enough pairwise tests are performed, we will with high probability find one or more "significant" differences. This problem is known as the *multiplicity effect* or *selection*

effect. Statistical procedures have been designed to take into account and properly control for the multiplicity effect; they are called *multiple comparison procedures.*

A crude approach to deal with the multiplicity effect is the Bonferroni[2] method, which rejects the pairwise null hypothesis $\theta_i - \theta_{i'} = 0$ when the p-value is less than α/k', where α is the preset FWE level and k' is the number of tests. According to the Bonferroni method, p-values obtained by single pairwise tests are adjusted to $\tilde{p} = \min(k'p, 1)$. This method neglects the possible dependency between the p-values of different pairwise difference tests.

Very closely related to the Bonferroni method is the *Šidàk method* [WY93], which rejects the null hypothesis H_i when the p-value p_i is less than $(1 - (1 - \alpha)^{1/k'})$. This results in the Šidàk adjusted p-value $\tilde{p}_i = 1 - (1 - p_i)^k$. The Šidàk adjustments usually are less conservative, compared to the Bonferroni adjustments [WY93].

There are many alternative tests, ranging from slight adjustments of the Bonferroni method to very sophisticated techniques. The existing comparison procedures can roughly be categorised as analytical [HT87] or resampling based [WY93]. The former approaches require certain distributional assumptions of the underlying statistical model, and typically use table lookup to make a probability statement. The latter approaches generate empirical distributions of the relevant statistics by resampling from the data set at hand, thereby removing the risk of making false statements due to unsatisfied assumptions. Evidently, the resampling approach involves much more computation than the analytical approach.

3.5.3. Multiple Comparison Procedures

The characteristics of a particular experimental design often prescribe adjustments to general tests for differences or they make special purpose tests necessary. The experimental design that captures the subject of this study is the *one-way repeated measures design*, which is displayed in Table 3.1. In such designs, blocks consisting of a random sample of, say, n experimental units drawn from a large population constitute the random factor. Each unit is measured under k different conditions. The conditions of measurements are fixed in advance, and constitute the treatment factor. In the terminology of this study, experimental units correspond to the observations from the test set, and the treatment factor corresponds with the regression or classification model type.

The general setting of this section is Table 3.1 (the one-way repeated measures design with k different prediction models which predict the observations from the *same* random test set of size n). The deviation of the predicted value from the true value is assumed to be measured as squared error, but any other error measure could be used equally well (e.g., absolute error).

[2]This method originally due to R.A. Fisher ([Fis35]) is popularly known as the Bonferroni method since it uses the Bonferroni inequality (which says that the probability of a union of events is less than the sum of the individual event probabilities).

	Functions						
Observations	f_1	f_2	...	f_i	...	f_k	Total
1	Y_{11}	Y_{12}	...	Y_{1i}	...	Y_{1k}	$Y_{1.}$
2	Y_{21}	Y_{22}	...	Y_{2i}	...	Y_{2k}	$Y_{2.}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
j	Y_{j1}	Y_{j2}	...	Y_{ji}	...	Y_{jk}	$Y_{j.}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	Y_{n1}	Y_{n2}	...	Y_{ni}	...	Y_{nk}	$Y_{n.}$
Total	$Y_{.1}$	$Y_{.2}$	...	$Y_{.i}$	...	$Y_{.k}$	
Means	$\overline{Y}_{.1}$	$\overline{Y}_{.2}$	...	$\overline{Y}_{.i}$	...	$\overline{Y}_{.k}$	

Table 3.1: One-way repeated measures lay-out.

When the observations are not randomly drawn from a population, but result from a (highly) autocorrelated time series, the subsequent approach seems not to be justified. Diebold and Mariano [DN90] discuss the comparison of predictive accuracy of two time series models; they leave the multiple comparison problem for further research.

Let $\mathbf{Y}_j = (Y_{j1}, Y_{j2}, \ldots, Y_{jk})$ denote the vector of prediction errors for the jth observation $(1 \leq j \leq n)$. The following model is assumed:

$$\mathbf{Y}_j = \mathbf{M}_j + \mathbf{E}_j \ \ (1 \leq j \leq n), \tag{3.14}$$

where all the $\mathbf{M}_j = (M_{j1}, M_{j2}, \ldots, M_{jk})$ and $\mathbf{E}_j = (E_{j1}, E_{j2}, \ldots, E_{jk})$ are distributed independently of each other as k-variate normal vectors, the former with mean vector $\boldsymbol{\theta}$=$(\theta_1, \theta_2, \ldots, \theta_k)$ (the vector of model effects) and variance-covariance matrix $\boldsymbol{\Sigma}_0$, and the latter with mean vector $\mathbf{0}$ and variance-covariance matrix $\sigma^2\mathbf{I}$. Thus, the $\mathbf{Y}_j$'s are independent and identically distributed (i.i.d.) $N(\boldsymbol{\theta}, \Sigma)$ random vectors with $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_0 + \sigma^2\mathbf{I}$.

Exact procedures for making pairwise comparisons among the θ_i's can be constructed, if we impose special restrictions on the form of $\boldsymbol{\Sigma}$. The least restrictive of such models is the *spherical model*, which assumes that all pairwise differences of the sample means of the regression models have the same variance (for more details see [HT87, CH90, WBM91, Hay88]). In practice, however, this assumption will rarely be satisfied [HT87, Hay88].

Therefore, Hochberg and Tamhane [HT87, page 215] propose a test, in case one the sphericity assumption may not hold. They propose the following approximate $100(1-\alpha)\%$ simultaneous

confidence intervals for the pairwise differences $\theta_i - \theta_{i'}$:

$$\theta_i - \theta_{i'} \in \left[\overline{Y}_{.i} - \overline{Y}_{.i'} \pm |M|^{(\alpha)}_{k^*,n-1} \sqrt{\frac{S_{ii} + S_{i'i'} - 2S_{ii'}}{n}} \right] \quad (1 \leq i < i' \leq k), \tag{3.15}$$

where $|M|^{(\alpha)}_{k^*,n-1}$ is the upper α point of the Studentized maximum modulus distribution (see [HT87, Table 6]) with parameter $k^* = k(k-1)/2$ and degrees of freedom $n-1$; and where $S_{ii'}$, is the estimated (co)variance between $Y_{.i}$ and $Y_{.i'}$:

$$S_{ii'} = \frac{\sum_{j=1}^{n}(Y_{ji} - \overline{Y}_{.i})(Y_{ji'} - \overline{Y}_{.i'})}{n-1} \quad (1 \leq i, i' \leq k). \tag{3.16}$$

The Studentized maximum modulus distribution is defined as the distribution of

$$\max_{1 \leq i \leq k} |T_i|,$$

where $|T_i|$ is the modulus of the k-variate t-distribution with ν degrees of freedom and common correlation of zero.

When the assumption of normally distributed Y_i-values is not justified in practice, the resampling method proposed by Westfall and Young [WY93, Algorithm 4.3] is recommended. Their resampling method departs from the same experimental model (3.14) as we have used; see [WY93] for a detailed description.

In this section we proposed a first step towards a sound methodology for performing and analysing studies that compare the predictive accuracies of several regression functions. Rather than providing a mere ranking, hypothesis testing should be used to determine whether a significant difference among functions has been found. The formal methods for the appropriate hypothesis tests originate primarily from the field of experimental design. We selected some of these formal methods, and showed their relevance to the type of study that is encountered frequently in the recent AI and Machine Learning literature. Although the general difficulties induced by the multiplicity effect and by the dependency among observations are easy to grasp, finding "the right" testing procedure is much more difficult. The literature on the subject is somewhat ambiguous, and requires a rather high entrance level of statistical knowledge, which AI-researchers do not always possess. This may explain why comparative experiments are often performed in a rather casual way in the AI and Machine Learning literature.

3.6. Conclusions

Feed-forward neural networks were the subject of this chapter. We addressed the graphical and mathematical representations of feed-forward neural networks, and showed the conceptual

resemblance of neural network learning to nonlinear regression. When neural networks are viewed as special purpose nonlinear regression methods, they can be studied as any other statistical method. In case the neural network topology is fixed, neural networks can be analysed statistically in manner similar to that in which parametric nonlinear regression models are analysed. However, the numerous network weights which generally are estimated, make statistical testing questionable. Additionally, the distribution of the weights in small samples is not known, so limiting distributions have to be used for the construction of confidence intervals, prediction intervals, and so forth. The question remains how useful these limiting distributions are for small samples.

We also paid attention to issues that are induced by having small samples. The generalisation issue, which directly stems from the bias/variance dilemma, addressed in the previous chapter, was described for neural networks in particular. Further, we discussed weight decay as a remedy for bad generalisation performance. The Bayesian perspective on neural network learning and prediction and the Bayesian interpretation of weight decay were addressed in Appendix A, which complements this chapter. In Bayesian terminology the weight decay term, which is added to the squared error loss function, implements a prior distribution on the weights.

Comparing the prediction performances of more than two different models in a statistically sound way is not so easy. The main issue is to incorporate the multiplicity effect in the statistical analyses. We reviewed some statistical multiple comparison procedures that are especially useful for AI-researchers, who often compare the performance of "their" method with the performances of some "rival" methods on the same hold-out set.

In the next chapter, which deals with the practical aspects of applying neural networks, generalisation and weight decay will receive more attention.

Chapter 4

Practical Aspects of Neural Networks

4.1. Introduction

In the previous chapter we introduced neural networks, and discussed them in a statistical framework. We indicated the conceptual resemblance of neural network learning to statistical nonlinear regression. When NNs are applied in practice, one meets many difficulties and one has many decisions to take. The specification of a neural network involves not only a selection of the inputs; but also the selection of the various components of a network, such as which type of network to use, which squashing function, which error criterion, which learning algorithm, how many hidden layers, and how many hidden units per layer. Once these network components have been specified, the NN has to be confronted with the data. The issue then becomes whether preprocessed data should be preferred to raw data; if so, how should the preprocessing be performed? Training neural networks results in an approximating function, which often suffers from two main difficulties. First, when no precautions are taken, NNs will overfit the data. Second, numerous local optima (in terms of the error criterion) will likely be found for a particular data set.

These practical issues are important factors which determine the success of neural network applications; therefore they require careful investigation. The impact of particular choices of the network components is largest in small sample problems, where statistical theory is not of much help. Monte Carlo simulations are required to examine the effect of particular choices. The practice of neural networks, consequently, is characterised by heuristic rules more than by firm theories. Applying NNs is often said to be more an art than a science.

The aim of this chapter is to lay down the choices concerning the different aspects of neural network modelling, and to establish a general network construction procedure. We address the way in which we arrive at a network solution for practical data modelling problems.

The outline of this chapter is as follows. Section 2 describes the (strained) relation between

the neural network field and statistics. Section 3 clarifies some of the often heard neural network myths. Section 4 briefly discusses the role software plays in the dissemination of NNs. Section 5 elaborates on the choices of main neural network components. Section 6 deals with the major difficulty NNs meet in practice: overfitting. Section 7 refines the general cross-validation procedure from Chapter 2 for the choice of neural network parameters. Section 8 presents the results of a simulation experiment designed to show the effects of weight decay on overfitting and on the number of local minima. Section 9 establishes the neural network construction procedure. Section 10 concludes the chapter.

4.2. Neural Networks Versus Statistical Models

The growing attention neural networks receive as a tool for data analysis, which statisticians traditionally considered to be their field of expertise, is a thorn in their flesh. Several statisticians ([Rip93a, Whi89b, Sar94]) have pointed at the similarity between neural networks and well developed statistical techniques. Statisticians have gained great expertise in data analysis, which goes far beyond linear regression, which some people presuppose as being "state-of-the-art" statistics: "In essence, in terms of its everyday practice, there has only been modest progress in regression analysis since the days of Gauss. Neurocomputing is now providing a breath of fresh air to this 200 year old subject." [HN90, page 121].

To a certain extent, the popularity of neural networks when compared with statistical methods may be caused by the failure of statisticians to communicate their methodologies and algorithms to non-statisticians. The vast amount of accumulated statistical knowledge puts up a barrier for consumers of their methods. Neural networks, on the other hand, are in an embryonic phase, which means that the accumulated knowledge is relatively small. The language used within the neural network community is another factor which may explain the success of the neural network. Due to their diversity in scientific backgrounds, neural network engineers have developed a universal language. The appealing terminology facilitates the propagation of neural networks to the 'outside world'.

The core problems of data analysis do not change when the techniques they are approached with are changed. Therefore, difficulties statisticians have run into will also affect neural network scientists. The general philosophies that underlie several statistical methods of data analysis have been (partially) reinvented by neural network scientists [Sar94]. As we have seen already, statistical projection pursuit and the feed-forward single layer NN are conceptually very close. The main difference is their popularity.

The great advantage of neural networks is the ability to capture many modern statistical methods into a single framework. Using the 'neuron' as a building block, many statistical models can be constructed, simply by collecting them in several layers and by interconnecting

them. Sarle [Sar94] illuminately describes the mapping of several statistical methods (such as PPR and principal components analysis) onto a corresponding neural network design. In [VD94], we explicitly show the convenience of the neural network concept when implementing projection pursuit regression.

It is clear that statisticians have much to say about model building, model diagnosis, model comparisons, and so forth. It is important, however, that they expose the statistically important issues to the neural network researchers in a clear way, so they can benefit from them.

4.3. Neural Network Myths

In the neural network literature, especially in the early part, some myths and half-truths have been promulgated [Sar94]. Sarle [Sar94] mentions the following myths: NNs are intelligent, NNs generalise, NNs are fault tolerant, and local optima are rare. Another frequently heard myth is that NNs are robust against noisy or incomplete data.

Few things can be said for the foregoing. There is no more intelligence in neural networks than in any other statistical method for data analysis. We have seen that NN learning is no more and no less than a form of nonlinear regression. The latter statement, of course, concerns only those neural networks that are employed and designed for problems of data analysis and not for understanding human learning processes.

The ability of NNs to generalise is similar to that of other statistical models. In this respect, it is important to distinguish between interpolation and extrapolation [Sar94]. Given sufficient and well dispersed data, it is possible for NNs to interpolate a sufficiently smooth function quite well. Extrapolation with NNs, however, is much more fault prone than in the linear case (in which extrapolation is also known to be risky). The statement "neural network generalise well" clearly requires differentiation; in most cases neural networks interpolate well, but there is no obvious reason why they should extrapolate well (see [GBD92]).

Fault tolerance is the ability to produce approximately correct outputs, even when some neurons malfunction. Networks with a large number of neurons with local effects are inherently fault tolerant; the effect of a single neuron on the total output is small. On the other hand, NNs are often ill-conditioned, that is, they have too many degrees of freedom compared to the number of training observations. In an ill-conditioned network small errors in the input data can give rise to strange outputs [Sar94].

Weight estimation is in essence a nonlinear optimisation problem. A typical characteristic of nonlinear optimisation problems is the presence of local minima. In the neural network literature, the possible occurrence of local minima is often neglected. That local optima do frequently occur is recognised in ([Rip93a, Sar94, GFR94]). In his discussion of the paper [Rip94], Breiman states:"There are other aspects of neural nets that puzzle me. For instance,

almost none of the neural net people seem to worry about landing in local minima. But it worries me. Is it a problem, and, if not, why not?". In his reply Ripley answers: "As Professor Breiman guesses, local minima *are* a problem, much ignored. I do my optimization carefully, including checking if I have reached a local minimum by checking the Hessian, and using many (often hundreds of) starting points." We also recognise local minima as a problem and try to find a good local optimum by making multiple restarts with random initial weights. When many neural networks have to be trained, the number of restarts, however, has to be restricted for computational reasons.

The statement that NNs are robust against noisy or incomplete data is justified for a pattern recognition task, in which the input signal is represented by a bitmap –a large grid of zeros and ones. The information is distributed among the many individual cells, and a single cell has almost no effect on the final classification. In a regression (data analysis) context, however, each method –also neural networks– is sensitive to incomplete or noisy data items, especially when the particular item plays an important role in the determination of the outcome. The robustness against noisy and incomplete data is more a characteristic of the particular problem than of the method of analysis.

4.4. Software

Software is important for the dissemination of a new technology such as neural networks. Neural network algorithms can of course be coded in a general purpose programming language (third generation programming language) such as, for instance, C++ or PASCAL. Applied researchers, however, do usually not have the required programming skills or do not have the time to code a neural network algorithm themselves. Their interest is in the application of neural networks to a particular problem, not in the technical details or in the implementation of a neural network algorithm.

Today, many commercial and freeware software packages are available. Their specificity, flexibility, and extent vary a lot. Some packages are restricted to one specific network type, usually feed-forward, and to one particular learning algorithm, usually error back-propagation. Others present many different network types and several learning algorithms. Some mathematical and statistical packages, such as MATLAB, SAS, and SPLUS, also support neural network modelling. For statisticians, this removes the barrier to the use of NNs, and it makes the performance of comparative studies easier as well. A good overview of available neural network software is given in the FAQ (Frequently Asked Questions) of the internet news group `comp.ai.neural-nets`.

For our experiments, we use the statistical package SPLUS (for UNIX), which provides an interactive computing environment for graphical data analysis, statistics, and computational

programming. Its distinguishing features are: easily modified graphics and advanced statistical functions implementing the leading ideas in modern statistical research.

All neural network experiments we shall perform use the neural network S-function developed by Ripley. This S-code is publically available (by anonymous ftp from `markov.stats.ox.ac.uk (192.76.20.1)` in directory `pub/S`). It implements a standard feed-forward neural network with one hidden layer, no recurrent connections, and one output unit; the squashing functions of the hidden units are sigmoid (and cannot be changed), the squashing function of the output unit can be linear or sigmoid. Skip layer connections can be incorporated, and training with a weight decay penalty term added to the error criterion is provided as well. Estimation of the weights is done by a quasi-Newton general purpose optimiser (see [Nas90, Chapter 15]) with first derivatives calculated by the back-propagation algorithm. Unlike back-propagation, a quasi-Newton general purpose optimiser needs no *a priori* specification of learning parameters, such as learning rate and momentum term.

Although the S-function seems fairly restrictive in the eyes of a neural network engineer, it is powerful enough to perform the type of research we aim for. It offers the possibility of performing neural network regression, and to evaluate and compare its results with other techniques within the SPLUS computing environment. Hence, there is no need for repeatedly transforming the data and the results into different formats (such data transformations are often needed when working with different software packages simultaneously).

4.5. Neural Network Components

A neural network consists of several components, which have to be specified before training can start. The main components are the network type, the error function, the activation function, and the learning algorithm. No instant rules that prescribe the best component choices are present, and theory is often of little help. The following subsections discuss each component briefly.

4.5.1. Network Type

The most frequently used neural network type is a feed-forward neural network with one hidden layer (see Figure 3.1), for which the approximation theorem holds [Cyb89]. This network can be extended, for instance, by the addition of more hidden layers or by the inclusion of direct connections from inputs to output. Many different types of neural networks have been developed, such as, Kohonen maps and recurrent networks (see [HKP91] for an overview). We exclusively use feed-forward neural networks with a single hidden layer and skip-layer connections.

4.5.2. Activation Function

The activation function of a neuron transforms an incoming signal into an output signal (see also figure 3.2). Activation functions ϕ, also called transfer or "squashing" functions, should have a number of properties. The first requirement is that they are positive monotone between the values $-B$ and B (or between 0 and B) for each signal between $(-\infty, \infty)$, where bound B is usually chosen to be 1. This requirement is necessary only for transfer functions of *hidden* units. For use in back-propagation, it is also required that ϕ be differentiable and that ϕ' satisfies a simple differential equation, thus facilitating the evaluation of weight updates via the chain rule for partial derivatives. Two related functions, which possess these properties, are

$$\phi_\beta(z) = \frac{B}{1 + \exp(-2\beta z)} \tag{4.1}$$

and

$$\psi_\beta(z) = B \tanh \beta z = B \frac{\exp(\beta z) - \exp(-\beta z)}{\exp(\beta z) + \exp(-\beta z)} = B - \frac{2B}{1 + \exp(2\beta z)}, \tag{4.2}$$

where B denotes the bound and β denotes the slope of the activation function. These functions clearly are differentiable, and the derivatives are given by the differential equations

$$\phi'_\beta(z) = \frac{2\beta B \exp(-2\beta z)}{(1 + \exp(-2\beta z))^2} = 2\beta\phi_\beta(1 - B^{-1}\phi_\beta) \tag{4.3}$$

and

$$\psi'_\beta(z) = \frac{4\beta B \exp(2\beta z)}{(1 + \exp(2\beta z))^2} = \frac{2\beta}{-2B}(\psi_\beta - B)(\psi_\beta + B), \tag{4.4}$$

respectively.

The simplest activation function, which is often used as the transfer function for the output unit(s), is the linear activation function. We use standard logistic squashing functions ($B = 1$ and $\beta = 0.5$), which transform incoming signals into an 0-1 range, for the hidden units, and a linear activation function for the output unit.

4.5.3. Error Function

The most widely used error function (objective or cost function) in neural networks is the Squared Error Loss function (SEL). This function is usually chosen without discussion, but several researchers have posed that SEL is not the most intuitive ([Urb92]) and robust ([Ber90]) method available for regression problems. The main disadvantage of SEL is the large influence of outliers on the estimated underlying function, due to the quadratic term. In statistics robust regression techniques are proposed to avoid this problem. In the future these robust regression techniques should also be considered for neural network learning. Especially, when one has a

limited data set, robustness becomes important. Small data sets do occur very often in economic data analysis.

The suitability of a specific error function depends on the purpose of the data modelling activity. When the purpose is to model typical system behaviour, a robust method is preferable, since then the atypical values are not weighted too heavily; but when the objective is to model atypical system behaviour, OLS will perform better, due to its sensitivity to large deviations. In the neural network literature, SEL is the common standard for regression, whereas for classification cross-entropy (Kullback-Leibler distance) is the standard; the latter is defined by

$$E = \sum_p \left[y_p \log \frac{y_p}{\hat{y}_p} + (1 - y_p) \log \frac{1 - y_p}{1 - \hat{y}_p} \right].$$

Software packages usually do not support error functions other than least squares and cross-entropy. In this thesis we adopt the least squares criterion.

4.5.4. Learning Algorithm

Learning algorithms try to minimise the error function. In fact, neural network learning algorithms are nonlinear optimisation procedures. They differ from general purpose optimisation procedures in the way the optimisation is carried out. Learning algorithms, such as error back-propagation, perform computation on the network itself: the network architecture offers a convenient way to compute the gradient information necessary for minimising the error function.

Back-propagation is a gradient descent algorithm, which at each iteration makes a step into a descending direction of the error function. From mathematics it is known that gradient descent algorithms converge slowly [Sca85, section 3.2]. When a fixed learning rate is chosen, it is likely that they do not converge at all. Almost each textbook on neural networks (e.g., [HKP91, Fre94]) elaborates on back-propagation, so we leave out the details.

Many algorithms have been invented that speed up the back-propagation algorithm. The best known variant of back-propagation is Quickprop [Fah88]. Schiffman *et al.* [SJW92] performed an extensive study on the comparison of different learning algorithms. They found Quickprop and Cascade correlation [FL90] among the best.

In Ripley's neural network SPLUS-code, a general purpose quasi-Newton optimiser is used to minimise the error function. It uses second order information (the Hessian) to calculate the optimal 'step size'; the Hessian is used to check whether a minimum (not a maximum or a ridge) has been found. Alternative nonlinear optimisation methods are found in [Sca85], for instance.

4.6. Data Preprocessing

Theoretically, there is no reason to scale the inputs onto a fixed interval, but there is often a good practical reason. Which proceeds as follows. The initial weight values of a neural network are often randomly drawn within a small cube in the weight space. Assume that there are large deviations in the ranges of the data components. Particularly with small networks, the smallest and largest weights involved will often be determined by the ranges of the components of the data. Due to permutations of hidden units, the weights associated with all global minima lie between two concentric shells in the weight space. Especially in the case of small networks whith the ranges of the components of the data differing a lot, the space between the two concentric shells occupies only a small part, or none, of the volume of the cube with initial weights. Each learning algorithm takes many steps to 'move' the weights into the concentric-shell region. This is quite troublesome, when using activation functions (e.g. sigmoids) in which the derivative is very small when some weighted sum is large in magnitude and completely of the wrong sign.

A real reason to scale network inputs is when weight decay is used as regularisation method. For weight decay to be effective, it is necessary that the inputs be comparable with the signal coming from the hidden units, which lie between zero and one when the logistic transfer function is used. This is the only real reason to scale the inputs.

Thus, in practice, scaling is useful to avoid bad initial conditions, and necessary for weight decay to be effective. The simplest scaling technique is to scale all data onto a constant range, say, $[0, 1]$. But there are alternatives that could be used.

Actually, we employ the following procedure to scale the data series x. Calculate the 0.025 quantile, q_1, of $x - \overline{x}$ and the 0.975 quantile, q_2. Define $z = \max(\text{abs}(q_1), \text{abs}(q_2))$. The scaled data series $\tilde{x}$ is then calculated by

$$\tilde{x} = \frac{(x - \overline{x})}{2\,z} + 0.5.$$

In this way at least 95 percent of the scaled data lies within the $[0, 1]$ range. This rescaling makes the signal transferred by each input unit comparable with the outputs of internal units. The problem with scaling onto a fixed interval is that outliers in the data can force the data to be scaled onto a very narrow interval. Our scaling procedure allows an outlier to remain an outlier, but scales the rest of the data more appropriately.

4.7. Overfitting

In Chapter 3 (section 3.4) we have already noticed that a real concern with a modelling procedure as flexible as neural networks is that one might find considerable spurious structure in data

for which the signal-to-noise ratio is small. This false structure would reflect the sampling fluctuations in the noise and would provide a misleading indication of the association between the response and predictor variables. One would expect this effect to be especially severe for small samples in high dimensions.

4.7.1. An Example

Table 4.1 summarises the results of applying neural network learning to pure gaussian noise, that is, $g(\mathbf{x}) = \epsilon$ with $\epsilon \sim N(0, 1)$. Results are presented for three sample sizes ($n = 50, 100, 200$), three sizes of the hidden layer ($N_h = 2, 4, 10$), and five covariates randomly drawn from the interval [0,1]. The table gives the percentage of the points of the lower half of the distribution of the ratio of the sum of squared errors (SSE) on the training set ($\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$) and the SSE of the true model ($\sum_{i=1}^{n} y_i^2$). Ideally this ratio should be close to one, as the neural network should not fit to the noise. A value close to zero indicates that the neural network is actually fitting a model to the noise. As an example, in Table 4.1 an entry in the column with heading 10% indicates that 10% of all (100) ratio values are below the actual value of that specific entry.

Table 4.1: Modelling pure noise.

n	N_h	1%	5%	10%	25%	50%
50	2	0.33	0.41	0.44	0.50	0.59
	4	0.15	0.20	0.24	0.29	0.38
	10	0.00	0.00	0.01	0.02	0.03
100	2	0.58	0.64	0.67	0.71	0.76
	4	0.44	0.47	0.48	0.52	0.57
	10	0.13	0.14	0.16	0.22	0.26
200	2	0.76	0.79	0.80	0.83	0.86
	4	0.63	0.66	0.69	0.72	0.76
	10	0.40	0.42	0.43	0.48	0.53

Table 4.1 indicates that the neural network overfits the data severely, even when a hidden layer of size 2 is used. As expected, the degree of overfitting is large, when the sample size (n) is small; it decreases when the sample size increases. Even with 200 observations and only 2 hidden units, the neural network 'explains' a significant part of the variance; 50% of the ratios are less than 0.86. The experiment reveals that it is necessary to take precautions against overfitting; otherwise it is impossible to make good predictions.

4.7.2. Remedies

The remedy for overfitting is to control the complexity of the neural network. There are two main approaches: model selection and regularisation. The same approaches were used to reduce the negative consequences of multicollinearity in Chapter 1 (section 1.3.2).

Model selection for neural networks involves choosing the number of hidden units, the connections, and the inputs. Miller [Mil90] elaborates on model selection for linear regression. The neural network approach to model selection is *pruning*, i.e., start with a large network and remove connections or units during training by various algorithms ([lC89, WHR91, Ree93]). The statistical approach to model selection is to estimate the generalisation (prediction) error for each model and to choose the model with the minimum estimated error. For nonlinear models, the generalisation error is often estimated by cross-validation.

"Regularisation involves constraining or penalizing the solution of the estimation problem to improve generalization by smoothing the predictions" [Sar95]. Two common approaches to regularisation in neural networks are: stopped training and weight decay. The first is most popular among neural network users.

The simplest approach to stopped training is to stop training after a predetermined number of "epochs", which are complete presentations of the whole training set. It is obvious that this approach can only be suboptimal. A more realistic approach is to use a test set of data to indicate the error on 'unseen' cases; these data may not be used during training. When the error on the test set starts to increase, training is terminated. The idea is to prevent the network from overfitting the training data, so that a desirable degree of generalisation can be reached. To measure the degree of generalisation, a third independent set (the validation set) is necessary to estimate the out-of-sample performance of the network. Finnoff [FHZ93] compares the performances of different strategies to stopped training on various artificially created data sets.

In the previous chapter (see 3.4) we have already introduced weight decay, which prevents a neural network from overfitting by smoothing the resulting fit of the network. Sarle [Sar95] discusses several regularisation methods and presents the results of simulations investigating the differences between them. He finds that stopped training works well compared to weight decay *only* for the linear functions. Additionally, weight estimation remains a mathematically well defined optimisation problem in the case of learning with weight decay, whereas in the case of stopped training the actual optimisation problem is not well defined. We will use weight decay to regularise neural network solutions. The selection of the neural network model will be performed in the statistical way. Figure 4.1 illustrates the effect of weight decay in smoothing the fit to the data. The figure shows all observations from the total data set, the fit obtained with weight decay, and without weight decay. It is seen that even a small weight decay value effectively smooths the "bumpy" behaviour of the fit obtained when no weight decay parameter is added.

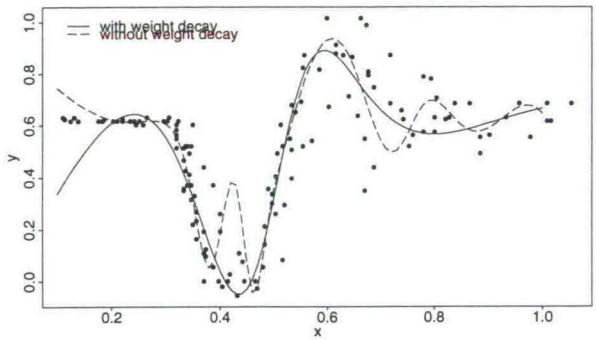

Figure 4.1: Feed-forward neural networks with 30 hidden units, trained on 50 observations from the motor cycle data set [Hae90] without weight decay; the other with a weight decay term of 0.0001.

Table 4.2 shows the effect of weight decay for the pure noise example. Comparing table 4.2 is compared with Table 4.1, shows that the values in the cells have increased (in the ideal case the value should be one). So weight decay effectively reduces the temptation of neural networks to overfit the data. A weight decay value of 0.1 was employed in constructing Table 4.2; a larger value would have had more effect.

Table 4.2: Modelling pure noise with weight decay of 0.1.

n	N_h	1%	5%	10%	25%	50%
50	2	0.61	0.63	0.65	0.73	0.79
	4	0.46	0.51	0.53	0.58	0.66
	10	0.39	0.43	0.48	0.56	0.67
100	2	0.76	0.80	0.81	0.85	0.88
	4	0.59	0.62	0.65	0.71	0.78
	10	0.46	0.50	0.54	0.61	0.69
200	2	0.87	0.88	0.90	0.92	0.94
	4	0.77	0.78	0.79	0.82	0.85
	10	0.61	0.67	0.69	0.72	0.78

4.8. Cross-validation for Neural Networks

In chapter 2 we outlined the use of cross-validation for the choice of parameters in non-parametric regression. In this section we discuss the peculiarities of cross-validation in the selection of neural network parameters. The flexibility of the approximating function constructed by a neural network depends on the number of hidden units, N_h, and the value of the weight decay parameter, λ. More hidden units absorb more degrees of freedom and a high value of the weight decay parameter prevents the weights from growing too large. Let θ denote both the flexibility parameters: number of hidden units N_h and weight decay value λ. The parameter values are set by minimising $CV(\theta)$ using data set $D = (\mathbf{x}_i, y_i)_1^n$.

Weiss [Wei91] states that leave-one-out cross-validation is computationally demanding and results in a high variance estimator of the prediction accuracy in small samples. The computational burden is even larger when cross-validation is used for parameter choice in neural networks, because n neural networks have to be constructed for a data sample of size n. Zhang [Zha93] and Kohavi [Koh95] illustrate that k-fold cross-validation, in which 5 or 10 equally sized parts of the data are left out, performs well in selecting models. In the remainder we will use k-fold cross-validation with k equal to 5 or 10 (see also 2.4). The choice between 5- or 10-fold cross-validation is based upon the expected total computation load.

Moody and Utans [MU94] propose the following refinement to the general cross-validation procedure (described in section 2.4) to make it usable for neural networks. Train a neural network on the whole data set D to a good solution $\mathbf{w}^*$. Permute the data set D randomly and decompose it into k mutually exclusive subsets S_i of roughly equal size, where $i = 1, \ldots, k$. Construct the cross-validation sum of squares of the trained network $f_\theta(\mathbf{x}, \mathbf{w}^*)$

$$CV(\theta; \mathbf{w}^*) = 1/k \sum_{i=1}^{k} \left\{ \frac{1}{|S_i|} \sum_{p \in S_i} \left(f_\theta(\mathbf{x}_p, \mathbf{w}; \mathbf{w}_0 = \mathbf{w}^*, D^{-i}) - y_p \right)^2 \right\}, \qquad (4.5)$$

where D^{-i} denotes data set $D \setminus S_i$ and $f_\theta(\mathbf{x}_p, \mathbf{w}; \mathbf{w}_0 = \mathbf{w}^*, D)$ denotes the output of the neural network with fixed parameters θ trained on data set D from starting weights $\mathbf{w}^*$ when faced with input $\mathbf{x}_p$.

Notice that we explicitly made CV dependent on the (locally) 'optimal' weight vector $\mathbf{w}^*$. Inside this cross-validation procedure, each neural network is trained from starting weights $\mathbf{w}^*$ after a subset S_i is removed from the training data D. This perturbs the 'optimal' weights to obtain new weights, which are assumed to be relatively close to the locally-optimal weights. Under this assumption, the error computed for the 'perturbed models' thus estimates the prediction error for the model with locally optimal weights $\mathbf{w}^*$ [MU94]. This assures that the prediction error corresponding to the network $f_\theta(\mathbf{x}, \mathbf{w}^*)$ is estimated. The proposed cross-validation procedure is illustrated in Figure 4.2. With *random* starting weights, the network could converge to

a minimum weight vector different from the one corresponding to $\mathbf{w}^*$, which would correspond to a different NN model. It would be unclear which network's prediction error has actually been estimated [MU94].

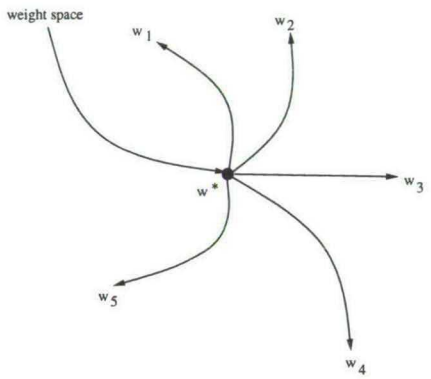

Figure 4.2: 5-fold cross-validation for neural networks (from [MU94])

When the neural network (with weights $\mathbf{w}^*$) overfits the complete data set so heavily that the squared error equals zero, the foregoing cross-validation procedure breaks down; leaving out part of the data will not result in weights different from $\mathbf{w}^*$, since the squared error was zero already. Although this seems a pathological case, we have had similar experiences in some practical applications; highly flexible neural networks fitting the training data very well, provided very low cross-validation errors, which turned out to be false after checking the final network by an independent set of data.

We will apply the neural network cross-validation procedure as proposed by Moody and Utans [MU94], but with some reservation. If we observe signs of severe overfitting and suspect the cross-validation error estimates to be faulty, we additionally apply a slightly different version of the cross-validation procedure. While searching for a good local minimum on the complete data set D, we keep the starting weights corresponding to the best weight vector found so far. This gives an "optimal starting weight vector" $\mathbf{w}_0^*$. In the cross-validation procedure, the neural networks are then trained from this starting weight vector $\mathbf{w}_0^*$ instead of from $\mathbf{w}^*$. Consequently, in the equation (4.5) $\mathbf{w}^*$ is replaced by $\mathbf{w}_0^*$. In this way the cross-validation procedure handles severe overfitting situations appropriately.

Although the cross-validation procedure is much applied, we still are far from a complete understanding of its properties in nonlinear model selection: "My impression is that the use of cross-validation ideas in these non-linear and highly parametrised problems is not fully understood" [Rip94]. This clearly is an issue for further research. There further is a need for a

careful comparison of cross-validation with alternative measures of generalisation ability, such that researchers can employ the best generalisation measure in building their models.

4.9. Some Experiments

The experiments in this section illustrate that unlike what is generally assumed by many users of neural networks, many locally optimal weight vectors with varying prediction performances may be found for a particular learning problem. The experiments further show that weight decay not only reduces the degree of overfitting, but also reduces the *number* of different locally optimal weight vectors. Moreover, the cross-validation error provides a reasonable estimate of the true prediction error.

The first experiment is carried out as follows. First n(=10,000) 3-dimensional covariate vectors were (uniformly) generated from the $[0, 1]$ interval. Then, the corresponding response variables were computed from [1]

$$y_i = 0.1 \exp^{4x_{1i}} + 4/[1 + \exp^{-20(x_{2i}-0.5)}] + 3x_{3i} + \epsilon_i, \quad 1 \leq i \leq n, \tag{4.6}$$

with the ϵ randomly generated from a normal distribution with zero mean and standard deviation such that the signal-to-noise ratio equals 3. The signal-to-noise ratio s is defined as the standard deviation of the signal divided by the standard deviation of the noise. The fraction of the total variance of the response variable that is accounted for by the true underlying function is given by $1/(1 + 1/s^2)$. A signal-to-noise ratio of 1, for example, means that 50% of the total variance is explained for by the true underlying function; an s of 2 means 80% is explained for; an s of 3 means 90% is explained for. The data in the complete sample are rescaled such that 95% of each data component lies within the $[0, 1]$ range. From the 10,000 scaled (x_1, x_2, x_3, y) vectors, a random sample of size 100 is drawn, which is used for training the neural network.

A neural network is repeatedly fitted to the 100 sample points, each time with different (randomly chosen) initial weight vectors; fifty repetitions are made. The neural network consists of 6 hidden units, a weight decay term, a linear output unit, and skip layer connections from input to output. The weight decay parameter takes the values 0.0001, 0.001, 0.01, and 0.1, respectively.

To measure the prediction performance of the neural network, the following approximation to the (scaled) predictive-squared error PSE is used

$$\text{PSE} = \sum_{i=1}^{n} [y_i - f(\mathbf{x}_i, \mathbf{w})]^2 / \sum_{i=1}^{n} [y_i - E(Y)]^2, \tag{4.7}$$

[1] The data generating function is an adjusted version of the one used in [Fri91].

where PSE is calculated by the remaining 9,900 artificially generated data vectors, so n equals 9,900. Since the data generating function consists of only three covariate variables, the estimation of the prediction error will be accurate. In practice, however, such an accurate estimate is difficult to obtain, because of limited data sets. We have seen that cross-validation has been designed to provide reliable estimates of the PSE in the case of small samples. Therefore, the 5-fold cross-validation error estimate, which is denoted by CV, is calculated as well.

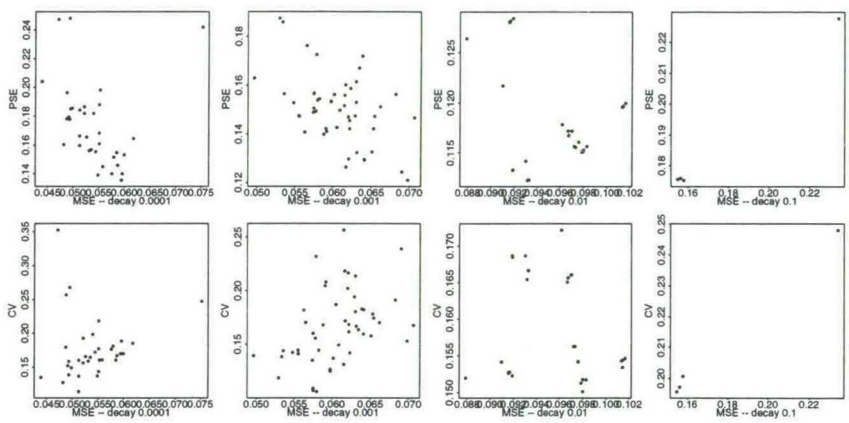

Figure 4.3: PSE vs. MSE and CV vs. MSE (after 50 restarts) by varying decay λ; data generating function (4.6) with a signal-to-noise ratio of 3.

Figure 4.3 shows the results of the experiment with a signal-to-noise ratio of 3. A signal-to-noise ratio of 3 implies a theoretical PSE of 0.1. The first row of "windows" in Figure 4.3 displays the PSE calculated by (4.7) against the (in-sample) MSE for each neural network resulting from the fifty random restarts of the training process; the neural network contains 6 hidden units and a weight decay value of 0.0001, 0.001,0.01, and 0.1, respectively. The second row displays the same information for the cross-validation estimate of the prediction error CV. The values corresponding to a linear model fitted to the data by OLS are: MSE 0.23, PSE 0.23, and CV 0.25.

When the weight decay term equals 0.0001, the different neural networks (with locally optimal weights), found after fifty random restarts, show a large spread in prediction performance PSE; some are even worse than the PSE of the linear model. The corresponding in-sample MSE indicates that the training data are overfitted; theoretically, MSE equals 0.1. Increasing the weight decay value to 0.01 makes the in-sample fit (MSE) less accurate, but improves the prediction performance PSE of the resulting neural networks; the PSE is not far away from the theoretical bound of 0.1. Further increasing the weight decay value to 0.1 constrains the fit so much that the prediction performance decreases.

Figure 4.3 further shows that weight decay influences the number of local minima neural networks fall into: when the weight decay parameter has a small value, there are many local minima, indicated by the numerous dots in each window. Increasing the weight decay parameter decreases the number of local optima found –the local minima become more and more concentrated onto a few isolated points: small weights make the signal entering a hidden unit fall into the (almost) linear part of the sigmoid transfer function. In this case, network inputs linearly affect the network output. It can be shown that when the least squares error function is used to determine optimal weights, there is a unique optimum (ruling out permutations). A large weight decay value causes weights to remain small. Weight decay, therefore, indirectly determines the number of local minima. However, one should not throw away the nonlinear modelling features of a NN by taking too large a weight decay parameter. Figure 4.3 also shows that a low in-sample MSE does not always correspond to a low PSE. Thus, the in-sample MSE should not be used as estimator of the PSE.

When we look at the pattern of cross-validation errors, displayed in the second row in Figure 4.3, we notice that it looks similar to the calculated PSEs. So, in practice the cross-validation estimates of the prediction errors are sufficiently accurate to base decisions concerning network parameter selection on. Based on the CV-information we would select a weight decay value of 0.01, which we would also have selected, had we the PSE-information at our disposal.

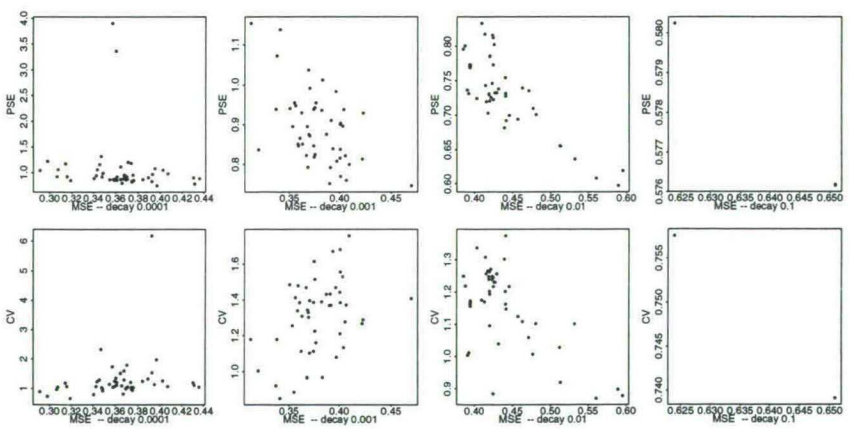

Figure 4.4: Different local minima (after 50 restarts) by varying sizes of λ; data generating function (4.6) with signal-to-noise ratio of 1.

The foregoing experiment is repeated with a signal-to-noise ratio equal to 1. So, in contrast with the previous experiment the signal is weakened; signal and noise account for the same amount of variation. The theoretical values for MSE and PSE are both 0.5. The results are

displayed in Figure 4.4. A linear model fitted to this new data sample achieves an in-sample MSE of 0.65, a PSE (calculated on 9,900 points) of 0.58, and a CVE of 0.74.

Figure 4.4 indicates that severe overfitting occurs for small values of the weight decay parameter, and that the prediction performance becomes very bad compared to the PSE of the linear model and the theoretical PSE of 0.5. Increasing the weight decay value to 0.1 makes the performance of the neural network solution resemble the performance of the linear model.

We conclude that when the signal is strong compared to the noise, the flexible neural network is able to find a solution that improves upon a parametric model, which is more biased in general. In case the signal is weak compared to the noise, it becomes much more difficult to outperform the biased parametric model: a large weight decay value is necessary to prevent the flexible neural network from performing worse than the parametric model.

4.10. The Network Construction Procedure

Iterative network construction methods have been developed in literature; some examples are the Cascade correlation algorithm [FL90], the SNC algorithm [MU94], and the CLS+ algorithm [RABCK93]. These methods start from a simple initial network and iteratively add components that approximate the remaining part of the signal (see [HKP91]). Projection pursuit regression follows a similar strategy. The alternative is to start with a large network and prune nodes, or to regularise the weights of a prespecified neural network by weight decay.

Resolving the issue of whether iterative network construction methods or using weight decay in a larger prespecified network is more effective, in general requires a systematic study.

We prefer the second alternative, namely to select the best regularised neural network on the basis of the k-fold cross-validation error. This approach has two advantages. First, when the weight decay term is added to the least squared error function, the learning problem remains a well defined mathematical optimisation problem. Second, the need for human interaction is minimal. The first alternative, iterative network construction methods, on the other hand, often require many subjective decisions by the user, and the complete problem solving process is mathematically less well defined.

This section lays down the procedure that is followed in the forthcoming chapters to arrive at a final neural network solution, i.e., a neural network with a specified architecture and with 'optimal' weights. The neural network construction procedure (NNCP) includes determination of the number of hidden units, the value of the weight decay parameter, and an optimal weight vector. The objective is to find a good solution within reasonable time. The NNCP starts with a rigid (not flexible) approximation to the data, and in a step-wise manner investigates whether adding more flexibility is justified by the data. We stress that the procedure is purely *heuristic*, but its clarity and modularity make it useful for practical use.

The transparency of our procedure enables researchers to better interpret and appraise its results. In time, hopefully, a generally accepted neural network methodology will arise. The difficulties in network construction are mainly caused by the multiplicity of locally optimal weights in small sample problems, and by the tendency of neural networks to overfit the data when too many hidden units are incorporated. We select the network parameters by cross-validation, which brings in some more difficulties; we mention the variance of cross-validation as an estimator of the prediction error, and the additional computational load.

The network construction procedure runs as follows. It assumes that the relevant inputs have been indicated, and the necessary data have been collected. The neural network includes skip-layer connections, i.e., direct connections from the input layer to the output layer. The two network parameters that are set by the procedure are the number of hidden units h and the weight decay parameter λ. A k-fold cross-validation is used to select both parameters. When data are abundant, it is preferable to use a randomly selected hold-out set instead of cross-validation, which will reduce the computation time considerably. The hidden units are selected from the set $\mathcal{H} = \{0, 2, 3, \ldots, h_{\max}\}$ where $h_{\max}$ denotes the maximum number of hidden units. The procedure starts with zero hidden units, which corresponds to the linear model.

The weight decay parameter is selected from the finite set $\Lambda = \{\lambda_{\max}, \ldots, \lambda_{\min}\}$ where $\lambda_{\min}$ and $\lambda_{\max}$ denote the minimum and maximum weight decay value respectively. Usually we take $\lambda_{\min} = 0$ and $\lambda_{\max} = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \hat{y}_i^{ols})^2$, but any other user-defined bounds can be used. The specific choice for $\lambda_{\max}$ requires some explanation. In Appendix A was shown that according to Bayesian statistics the weight decay parameter is proportional to the fraction of the variance of the network residuals and the variance of the weights. For $\lambda_{\max}$ we could adopt the fraction of the variance of the residuals obtained from a linear model (estimated by OLS) and a subjectively chosen weight variance of one.

Figure 4.5 presents the network construction procedure in pseudo-code. The weight decay parameter is denoted by λ, the number of hidden units by h, the network weights by $\mathbf{w}$, and the cross-validation errors by CV and cv. $E'(\mathbf{w}, \mathbf{w}_0, \lambda, h)$ represents the error criterion $\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i, \mathbf{w}, \mathbf{w}_0, \lambda, h))^2 + \lambda |\mathbf{w}|^2$ where $f(\mathbf{x}_i, \mathbf{w}, \mathbf{w}_0, \lambda, h)$ represents the neural network (with h hidden units, weights $\mathbf{w}$, and weight decay parameter λ), trained from the initial weight vector $\mathbf{w}_0$. Execution of the cross-validation procedure, as outlined in section 4.8, is denoted by $CV(\mathbf{w}^*)$. Finally, "next(h)" is a function that takes the next element from the set of hidden units $\mathcal{H}$.

Initially, the neural network model comprises no hidden units and the weights are strongly penalised by λ, which corresponds to a linear model estimated by ridge regression (or penalised OLS). When λ is decreased to zero, the model corresponds to a linear model estimated by OLS.

Figure 4.5: The General Network Construction Procedure

Begin procedure Construct-network
{ $h \leftarrow 0, \lambda \leftarrow \lambda_{\max}, CV \leftarrow \infty, cv \leftarrow \infty$
while ($CV > 0$ and $h < h_{\max}$) do
{for $\lambda \in \Lambda$
{$E_0 \leftarrow \infty$
repeat MAXIT
{sample $\mathbf{w}_0$
$\tilde{\mathbf{w}} \leftarrow \min_{\mathbf{w}} E'(\mathbf{w}, \mathbf{w}_0, \lambda, h)$
if $E'(\tilde{\mathbf{w}}, \mathbf{w}_0, \lambda, h) < E_0$ then $\{E_0 \leftarrow E'(\tilde{\mathbf{w}}, \mathbf{w}_0, \lambda, h)$; $\mathbf{w}^* \leftarrow \tilde{\mathbf{w}}\}$
}
calculate $CV(\mathbf{w}^*)$
if $CV(\mathbf{w}^*) < cv$ then $\{cv \leftarrow CV(\mathbf{w}^*)$; $\lambda_h \leftarrow \lambda\}$
}
if $cv < CV$ then $\{h^* \leftarrow h; CV \leftarrow cv$; $h \leftarrow$ next(h)$\}$
else $CV \leftarrow 0$
}
return(h^*, λ_{h^*})
}
End procedure

Inside the while-loop hidden units are added to the network; inside the for-loop the weight decay parameter is lowered. Hidden units are added either until the cross-validation error corresponding to the best λ (resulting from the For-loop) becomes larger than the optimal cross-validation error CV corresponding to the previous network or when the maximum number of hidden units $h_{\max}$ has been reached. For a particular number of hidden units h the for-loop determines the weight decay parameter λ_h with lowest cross-validation error cv. The repeat-loop searches for a good locally optimal weight vector by minimising $E'(\mathbf{w}, \mathbf{w}_0, \lambda, h)$, employing MAXIT restarts with randomly sampled starting weights $\mathbf{w}_0$. The network construction procedure is ended by making CV equal to zero when no improvement is obtained (i.e., no smaller cross-validation error results). The procedure returns the best number of hidden units and the corresponding best weight decay value. Of course, it is possible to return the corresponding optimal network weights as well. To improve the clarity of exposition, we do not show this additional step in Figure 4.5.

The variance of k-fold cross-validation due to different random group divisions is substantial

in general (see [WL94]). Therefore, it is important to calculate the cross-validation error each time with *identical* subdivision of the data in groups, to avoid the risk of drawing conclusions induced by the variance of cross-validation rather than by the change of a particular network parameter. We can take this variance of cross-validation into account by repeatedly calculating the cross-validation error, each time with a different division in groups, and use the *average* cross-validation error in our network construction procedure. However, this would increase the computational burden enormously; moreover, its effect on the selection of network parameters is unclear.

The network construction procedure implicitly assumes that increasing the network's hidden layer does not result in a cross-validation error that first increases and then decreases, once a sufficiently large number of hidden units has been added. Our reasoning is that a larger network should be able to encompass a smaller network and should therefore improve over the smaller one when a more complex structure is supported by the data. If the data do not require a complex approximation, making the network even more complex makes no sense.

4.11. Conclusions

The popularity of neural networks relative to competing statistical techniques can be explained by the appealing appearance and better marketing rather than by their distinctive data modelling qualities. Statisticians, therefore, have a sceptic attitude towards neural networks.

This chapter mainly addressed the (subjective) choices investigators have to make when they apply neural networks to data modelling problems. We observed that overfitting constitutes a fundamental problem. It is a direct consequence of the bias/variance dilemma (section 2.3), which affects the whole class of model free regression methods. Weight decay was shown to be an effective remedy for the neural network's temptation to overfit the data.

We further observed that the occurrence of different locally optimal weights for a learning problem, is the rule rather than the exception. The presence of local optima hampers the design of automatic network construction algorithms. Although global optimisation methods are available, their use in empirical research is computationally still infeasible (at least on a SUN Sparc station 1). As an alternative to global optimisation, we employed a multi-start algorithm, which repeatedly trains a neural network from different starting weights, inside the network construction procedure.

The degree of subjectivity involved in building neural networks makes an explicit algorithmic representation of the process necessary, in order to effectively pass on empirical results to others. The network construction procedure that we developed mainly serves this purpose. When in the remainder neural networks are applied, it will be done according to this procedure.

Chapter 5

Neural Networks in Econometric Time Series Modelling

5.1. Introduction

In Chapter 1 we outline the general process of economic modelling. The issues that were discussed hold for both cross-sectional data and stationary time series data.

This chapter deals with the econometric modelling of time series, in particular the modelling of *nonstationary* time series. Nonstationary time series have always caused problems in their analysis. It has been recognised that standard significance tests are no longer valid, and that spurious relationships can be found. Cointegration analysis and error-correction models have been developed for modelling nonstationary time series; we will summarise its theory. This theory assumes the models to be linear in the variables.

Next, nonlinearities are incorporated in the cointegration analysis and in the error-correction mechanism. A first step towards nonlinear generalisations of cointegration and error-correction is taken, using the theory and practice of neural networks of the chapters 3 and 4.

The outline of this chapter is as follows. In section 2 we address some of the consequences of using time series models for prediction. Section 3 introduces the concepts of cointegration and error-correction models (ECM). Section 4 discusses *nonlinear* cointegration and ECM, and discusses their implications for the practice of economic modelling. Critical values for Dickey-Fuller tests on neural networks are derived. Section 5 concludes the chapter.

5.2. Time Series

In this section we explain in an intuitive way why time series modelling can be statistically more problematic than regression modelling with cross-section data)

In Chapter 1 we have already discriminated between stationary and nonstationary time series. Figure 5.1 and Figure 5.2 present extreme examples of a (white noise) stationary process and a (random walk) nonstationary process, generated by

$$X_t = X_{t-1} + \epsilon_t \; ; \; \epsilon_t \text{ i.d.d. } (0, \sigma^2).$$

For the random walk process, it is easily derived that $E(X_t) = 0$ (provided $x_0 = 0$) and $\text{Var}(X_t) = t\sigma^2$, which clearly does not meet the definition of stationarity (see Chapter 1).

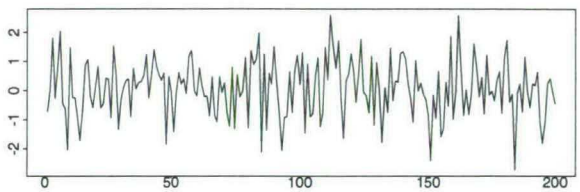

Figure 5.1: White noise process

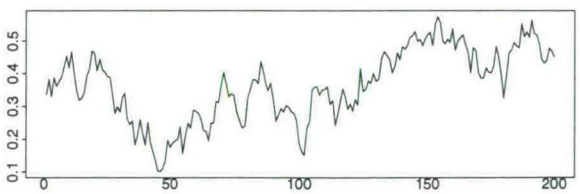

Figure 5.2: Random walk process

In classical regression, a first requirement is that the independent variables are nonstochastic or at least distributed around a constant mean and have a finite variance. Cross-section data normally have this feature. When the parameters in the model have been estimated on a particular data set, predicting new observations is generally a matter of interpolation. This statement extends to stationary time series, such as the white noise process.

In case the independent variables form a nonstationary time series, the series does not frequently return to its mean. Predicting new observations generally requires extrapolation rather than interpolation. Extrapolation is recognised as being risky for each regression technique, in particular for the flexible techniques. Assuming an irregular underlying model seems to exclude any sensible form of extrapolation. WHAT-IF analysis, which is often used to evaluate possible scenarios, typically extrapolates the model. In the light of the statements made above, this may result in erroneous conclusions.

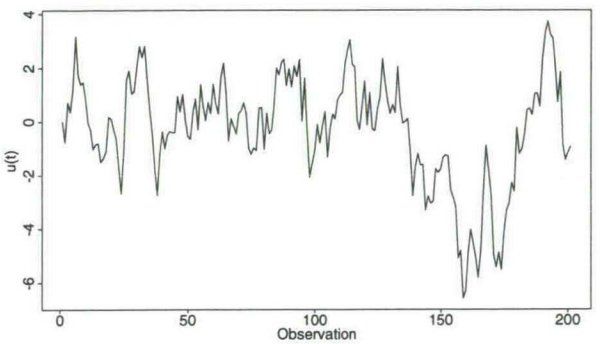

Figure 5.3: AR=0.9; MA=0.0

Between white noise and random walk processes, there is a whole range of classes of time series that are stationary but exhibit a certain degree of sluggishness in changing, for instance, the AR(0.9) process $x_t = 0.9x_{t-1} + \epsilon_t$, depicted in Figure 5.3. This sluggishness is statistically indicated by strong autocorrelation. In small data sets strong autocorrelation makes observations 'stick together' in variable space. The effect of including strongly autocorrelated stationary time series as predictor variables can be similar to that of nonstationary time series, i.e., prediction requires extrapolation rather than interpolation. To achieve the same level of information, more observations are needed on highly autocorrelated time series than on non-autocorrelated time series.

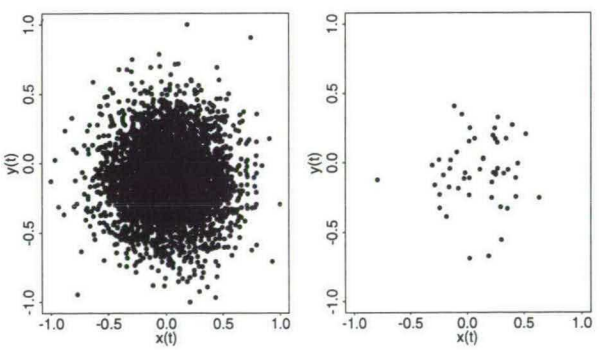

Figure 5.4: x_t and y_t are both white noise; the first plot shows 4000 samples, the second 50 samples.

Figures 5.4 and 5.5 show that autocorrelation in time series affects the dispersion of the

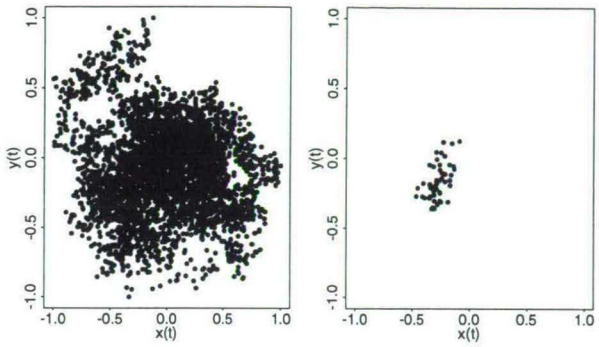

Figure 5.5: x_t and y_t are both AR processes with AR=0.99; the first plot shows 4000 samples, the second 50 samples.

training data in the variable space. If we assume that x_t and y_t are the relevant predictor variables of a particular phenomenon, then it is clear that the 50 observations in Figure 5.4 provide more information on the global shape of the relationship than the 50 observations do in Figure 5.5. The interpolation area in Figure 5.4 is larger than in that Figure 5.5. Macroeconomic time series, such as depicted in Figures 5.6 and 5.7, typically do have high autocorrelation.

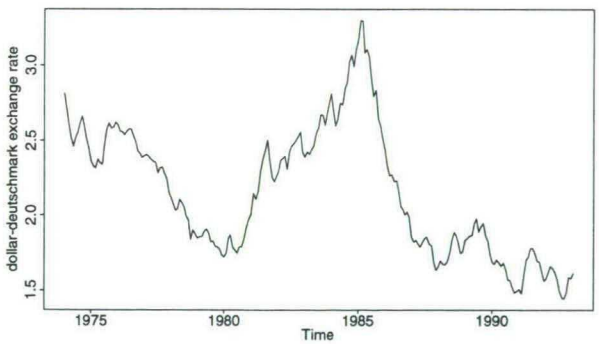

Figure 5.6: Time series representation of the dollar-deutschmark exchange rate (on monthly basis)

The foregoing illustrated the case of two explanatory variables and 50 observations. Economic relationships are often between more than two variables, and although we usually have some hundreds of observations, the dispersion problem becomes worse with each extra variable added. Therefore, models that include macroeconomic time series as predictor variables will

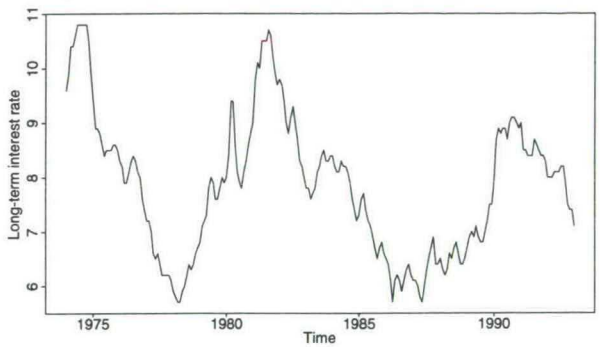

Figure 5.7: Time series representation of the German long-term interest rate (on monthly basis)

presumably have problems in achieving an acceptable level of long-run prediction accuracy. Hence, prediction requires extrapolation skills rather than interpolation skills, and extrapolation is generally recognised as being risky.

5.3. Cointegration and Error-correction

The previous section made a distinction between two different classes of time series: stationary and nonstationary. This section focusses on nonstationary time series and on its implications for statistical inference.

The nonstationarity of time series has always been regarded as a problem in econometric analysis. When modelling series which are subject to a deterministic or a stochastic trend, one is likely to end up with a model showing apparently promising diagnostic test statistics (high R^2 and significant t-values), even if regression analysis makes no sense [BDGH93]. Hendry [Hen93] gives an example of a model that explains inflation by a certain exogenous variable that meets all relevant statistical criteria, but which turned out to be the cumulative rainfall in the UK. This problem is known as the *spurious regression* problem.

Since almost all economic data series contain trends, it follows that these series have to be detrended before any sensible regression analysis can be performed. In the past a popular method that attempts to overcome the problem of spurious regression was to estimate the relationship between the rates of changes (differences) of variables rather than between absolute levels. Two problems, however, arise when concentrating attention on relationships among differenced variables. First, valuable information about the long-run relationship between the levels of variables, if present, will be lost. Second, if a long-run relationship in levels exists and

if its disturbance term is not autocorrelated, then the disturbance term of the model estimated in differences will be autocorrelated –in particular, it will have a simple moving average form. This will, consequently, influence the parameter estimates as indicated in Chapter 1 (section 1.3.4).

5.3.1. Dickey-Fuller Tests

Within the general class of nonstationary time series there is a large subclass that can be characterised by the order of *integration*: a nonstationary series X_t which can be transformed to a stationary series by differencing d times[1] ($\Delta^d X_t$) is said to be *integrated* of order d, conventionally denoted as $X_t \sim I(d)$.

Before any sensible regression analysis can be performed, it is essential to characterise the time series data by the order of integration, provided the variable can indeed be transformed into a stationary variable by differencing. Eyeballing the plot of the time series and inspecting the autocorrelation plot are two simple means that give a quick impression of the time series type. Time series can be characterised in a formal way by statistical hypothesis testing. Two appropriate tests are the unit root test due to Dickey and Fuller (DF test) and the augmented DF test (ADF test); see [BDGH93, Chapter 4] for details on these and other tests.

In unit root tests the null hypothesis $y_t \sim I(1)$ is tested against the alternative $y_t \sim I(0)$. The DF statistic tests for the restriction $\gamma_0 = 0$ in one of the following transformed equations

$$\Delta y_t = \gamma_0 y_{t-1} + \epsilon_t, \tag{5.1}$$

$$\Delta y_t = \gamma_0 y_{t-1} + \alpha + \epsilon_t, \tag{5.2}$$

$$\Delta y_t = \gamma_0 y_{t-1} + \alpha + \beta t + \epsilon_t, \tag{5.3}$$

implicitly assuming y_t is an AR(1) process. Which of the three equations should be used for testing is determined by the significance of the constant ($\hat{\alpha}$) and the trend ($\hat{\beta}$). Critical values on the t-values of γ_0 are tabulated in [Ful76].

If y_t is an arbitrary AR(p) process, then the disturbances ϵ_t in the foregoing equations will not be white noise, which causes the estimate of γ_0 to be inaccurate. To allow for arbitrary AR(p) processes in testing for unit roots, the DF test is augmented. The augmented DF test (ADF test) concerns the null hypothesis $y_t \sim I(1)$, that is, $\gamma_0 = 0$ in

$$\Delta y_t = \gamma_0 y_{t-1} + \sum_{i=1}^{p} \gamma_i \Delta y_{t-i} + \epsilon_t, \tag{5.4}$$

$$\Delta y_t = \gamma_0 y_{t-1} + \alpha + \sum_{i=1}^{p} \gamma_i \Delta y_{t-i} + \epsilon_t, \tag{5.5}$$

[1] For example, $\Delta X_t = X_t - X_{t-1}$ and $\Delta^2 X_t = \Delta(X_t - X_{t-1}) = X_t - 2X_{t-1} + X_{t-2}$

$$\Delta y_t = \gamma_0 y_{t-1} + \alpha + \beta t + \sum_{i=1}^{p} \gamma_i \Delta y_{t-i} + \epsilon_t, \tag{5.6}$$

where the inclusion of a constant or a trend is, again, based on the significance of the $\hat{\alpha}$ and $\hat{\beta}$. Holden and Perman [HP94] suggest a sequential procedure for unit root testing, which tests joint null hypotheses on the various parameters in a step by step manner. The number of lags p to include has to be large enough to ensure that the transformed regressions are well specified, i.e., the disturbances are white noise.

It is important to realise what the consequences are of selecting (5.4), (5.5), or (5.6). Equation (5.4) concerns the testing of the null hypothesis y_t is a random walk against the alternative hypothesis y_t is AR(p) with mean zero. Using equation (5.5), one tests the null hypothesis y_t is a random walk with drift (deterministic linear trend) against the alternative y_t is AR(p) with non-zero, but constant, mean. Using equation (5.6), one tests the null hypothesis y_t is a random walk around a nonlinear deterministic trend against the alternative y_t is AR(p) with a deterministic trend. If the latter alternative hypothesis is true, then y_t is called a trend stationary process. In practice, it is difficult to discriminate between a random walk with drift and a trend stationary process.

5.3.2. Testing for Cointegration

The desire to evaluate models which combine both short-run and long-run properties and which at the same time maintain stationarity in all variables, has prompted a reconsideration of the problem of regression using variables measured in levels. A requirement is *cointegration*, which is defined following [BDGH93, page 145] as follows:

Definition 1 *The components of the vector* $\mathbf{x}_t$ *are said to be co-integrated of order* d, b, *denoted* $\mathbf{x}_t \sim CI(d,b)$, *if (i)* $\mathbf{x}_t$ *is* $I(d)$ *and (ii) there exists a non-zero vector* $\boldsymbol{\alpha}$ *such that* $\boldsymbol{\alpha}^T \mathbf{x}_t \sim \boldsymbol{I}(\boldsymbol{d}-\boldsymbol{b})$, $\boldsymbol{d} \geq \boldsymbol{b} > \mathbf{o}$. *The vector* $\boldsymbol{\alpha}$ *is called the co-integrating vector.*

The most interesting case is when $d = b$. In practice, cointegration means that two series drift together instead of drifting apart; for a non-technical illustration of cointegration see [Mur94]. The simplest test for cointegration, proposed by Engle and Granger, tests for the existence of a unit root in the residuals of the static regression; this test and its alternatives are described in [BDGH93, par. 7.2].

The Engle-Granger test for cointegration proceeds as follows; for ease of exposition the case of two time series is taken. Suppose the time series x_t and y_t are both $I(1)$. The Dickey-Fuller (DF) and augmented Dickey-Fuller (ADF) tests will determine whether the residuals ϵ_t of the static regression

$$y_t = \alpha + \beta x_t + \epsilon_t \tag{5.7}$$

contain a unit root: if they do, x_t and y_t cannot be cointegrated. The application of the Engle-Granger cointegration test comprises the following steps. First, the parameters α and β are estimated. Second, the residuals $\hat{\epsilon}_t$ are calculated from $y_t - \hat{\alpha} - \hat{\beta}x_t$. The DF test then constructs a second regression of the first difference of the residuals on the lagged residual

$$\Delta\hat{\epsilon}_t = \gamma_0\hat{\epsilon}_{t-1} + \nu_t, \quad \nu_t \sim \text{i.i.d}(0, \sigma^2). \tag{5.8}$$

The null hypothesis is $\gamma_0 = 0$, that is, $\hat{\epsilon}_t$ contains a unit root; implying y_t and x_t are *not* cointegrated. The ADF test accounts for possible serial correlation in the residuals ν_t; it adds p lagged changes in the residuals to the former equation, which amounts to

$$\Delta\hat{\epsilon}_t = \gamma_0\hat{\epsilon}_{t-1} + \sum_{i=1}^{p} \gamma_i \Delta\hat{\epsilon}_{t-i} + \zeta_t, \quad \zeta_t \sim \text{i.i.d}(0, \sigma^2). \tag{5.9}$$

The t-statistics on the fitted γ_0 provide the DF and the ADF tests of a unit root in the residuals $\hat{\epsilon}_t$. No formal guidelines are available for the choice of p; in practice, p is taken to be the largest significant lag that assures the residuals ζ_t are white noise. Equations (5.8) and (5.9) are sometimes augmented by a constant, a trend, or both, each requiring its own set of critical values. The addition of a constant or a trend to (5.8) or (5.9) is equivalent to using model (5.7) with a constant or a trend included.

5.3.3. Constructing Critical Values

In [EY87] Engle and Yoo construct critical values for the DF and ADF tests for small data sets consisting of 50, 100, and 200 observations, in which the number of variables k range from 1 through 5. Engle and Yoo assumed that the data are generated by

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \nu_t, \quad \mathbf{x}_0 = 0, \quad \mathbf{x}_t' = (x_1, \ldots, x_k)_t, \tag{5.10}$$

with

$$\nu_t \sim IN(0, \sigma^2 I_k),$$

and that the co-integrating regression takes the form

$$x_{1t} = \alpha + \beta_2 x_{2t} + \beta_3 x_{3t} + \ldots + \beta_k x_{kt} + \epsilon_t. \tag{5.11}$$

They estimated this regression by OLS. The covariance matrix of the innovations ν_t is taken to be identity without any loss of generality, when assuming independent x_{it} series. The critical values of the DF and ADF tests are obtained as the t-statistics of $\hat{\gamma}_0$ in (5.8) and (5.9), respectively. These critical values were obtained through ten thousand replications. To examine

the movement of critical values in higher-order systems, Engle and Yoo generated data according to the following model

$$\mathbf{x}_t = \mathbf{x}_{t-1} + U_t, u_{it} = 0.8\, u_{it-1} + \nu_{it}, \;\; i = 1, \ldots, k, \tag{5.12}$$

assuming again $\nu_{it} \sim IN(0,1)$. They applied the ADF test with $p = 4$ to construct the critical values for this case. Engle and Yoo remarked that more variables are included in the transformed regression equations of residuals than necessary, which makes the reported critical values inefficient in a sense. In this way the critical values reflect the ignorance of the lag length in practice. The inefficiency will disappear in large samples [EY87]. Critical values can also be obtained from [Mac91], which provides an extensive source based on simulation experiments. The results are summarised by response surface regressions in which critical values depend on the sample size, the number of series, and whether or not a constant and a trend are included in the cointegrating relationship; asymptotic critical values can be read off directly.

5.3.4. Error-Correction Models

The fact that variables are cointegrated implies that there is some adjustment process which prevents the errors in the long-run relationship from becoming larger and larger. Cointegration is a necessary and sufficient condition to enable an error-correction model (ECM) formulation of the short-run dynamic model as represented by an ADL(p,q). This is formally represented in the Granger representation theorem [BDGH93, par. 5.3.1]. ECM currently represents the most common approach to situations in which it is desirable to incorporate both the economic theory on the long-run relationships among variables –also called equilibrium relationships or steady state relationships– and the short-run disequilibrium behaviour.

A simple example illustrates how an ECM looks like. Assume that y_t and x_t are $I(1)$ and cointegrated. So, there is a cointegrating regression $y_t = \alpha_0 + \alpha_1\, x_t + \epsilon_t$, and the residuals are stationary. The values of α_0 and α_1 can be either implied by economic theory, or estimated from the data. In the latter case the coefficients α_0 and α_1 can be estimated by a static regression or by a dynamic regression, in which the long-run coefficients are determined by substituting averages for every variable. In small samples the latter option is preferred [BDGH93, Chapter 7].

In practice, however, y_t and x_t will not often be in equilibrium, that is, $z_t = y_t - (\hat{\alpha}_0 + \hat{\alpha}_1\, x_t)$ will be non-zero in most cases. The discrepancy between y_t and $\hat{\alpha}_0 + \hat{\alpha}_1\, x_t$ is a measure of 'disequilibrium'. Since –by assumption– the system is in equilibrium in the long-run, the short-run process has to remove (at least partly) the disequilibrium. An error-correction model (ECM) implements this idea. For example, the simplest ECM for y_t would be

$$\Delta y_t = \zeta_1\, \Delta x_t + \gamma\, z_{t-1} + \nu_t, \tag{5.13}$$

where γ requires a negative sign to correct for the disequilibrium error of the previous period. Changes in y_t are explained by changes in x_t and by the disequilibrium error of the previous period.

5.4. Nonlinear Cointegration and Error Correction

The previous section (5.3) discussed the concept of cointegrating time series in a linear context. This relatively new concept has been the subject of many journal papers (At November 1994 our university's library contained over 220 articles that had cointegration in the title for the period after 1991) and books [BRe94, BDGH93]. However, not much has yet been written about nonlinear cointegration, which is an extension that comes to mind with the recent increase of interest in nonlinear modelling. Only a small group of researchers [Gra94, Sep94, MR91, GH91b] has paid attention to the nonlinear generalisation of the cointegration concept.

Our objective is to assess the usefulness of neural networks in testing for nonlinear cointegration. Sephton [Sep94] employed research on nonlinear cointegration tests on MARS (see Chapter 2). In his article, Sephton determines critical values for the ADF test on MARS, and finds some evidence for nonlinear cointegration in several selected cases. We follow a similar approach in constructing critical values for cointegration tests on residuals that result from a neural network regression.

5.4.1. The Characterisation of Time Series

To determine whether a time series is $I(1)$ or $I(0)$, a DF or ADF test is usually performed. Granger and Hallman [GH91b] show that standard DF and ADF tests reject the null hypothesis of $I(1)$ series too often, when the series is a nonlinear transformation f of a Gaussian random walk. For example, let

$$x_t = x_{t-1} + \epsilon_t, \quad \epsilon_t \equiv \text{i.i.d}(0, \sigma^2)$$

and

$$y_t = f(x_t).$$

Granger and Hallman propose to perform a DF or ADF test on the ranks of the series instead of on the original series, since the rank of a series is invariant to monotone transformations. The *rank* of a (finite) time series is defined as the rank of x_t among all observations, when ordered from low to high. Granger and Hallman [GH91b, table IV and V] give percentiles of the ranked Dickey-Fuller (RDF) and the ranked augmented Dickey-Fuller tests (RADF) under the null hypothesis that y_t is a monotone transformation of a random walk, for different sample sizes. It should be realised that there exist transformations of x_t that also escape from the RADF test. An additional investigation of the correlogram is, therefore, recommended.

Granger and Hallman further state that a nonlinearly transformed series generally cannot be cointegrated with the original series, e.g., in the above example y_t cannot be cointegrated linearly with x_t. This emphasises the importance of having the correct functional form when investigating a hypothesised long-run relationship between the two observed time series y_t and x_t. In fact, the foregoing justifies a nonlinear generalisation of the (linear) cointegration concept.

In [GT93], it is argued that the definitions of integratedness employed thus far are based on linearity. Therefore, they proposed to generalise the concepts of integratedness, and to distinguish between series that are long-memory in mean (LMM) and short-memory in mean (SMM), instead of $I(1)$ and $I(0)$ respectively. SMM is defined as follows.

Definition 2 *x_t is said to be SMM if*

$$\lim_h E[x_{t+h}|I_t] = D \tag{5.14}$$

where D is some random variable, and I_t denotes all available information available at time t. The case of most interest is where D is just a constant (the unconditional mean of x_t).

If $E[x_{t+h}|I_t]$ continues to depend on I_t as h increases, x_t is LMM. The definitions allow past information I_t to be used in a nonlinear way. In principle, D can include limit cycles and processes with strange (chaotic) attractors [Pet91] as well. These concepts, however, are not easily associated with the simple concept of equilibrium. Therefore, following [GH91a], we exclude them in what follows.

There are two more concepts, namely short- and long-memory in distribution. However, they are merely theoretical, and seem of little practical use. We, therefore, refrain from giving the definitions; interested readers are referred to [GT93].

Although the generalisation of the linear concept of integratedness theoretically makes sense, statistical tests that explicitly test whether a time series is SMM or LMM are not available (to the best of our knowledge).

5.4.2. Nonlinear Attractors

The concept of (linear) cointegration can be generalised in a nonlinear way [GH91a, Sep94]. In [GH91a], ACE (Alternating Conditional Expectations), which is well described in [HT90], is employed to identify nonlinear cointegration between two integrated time series. Granger and Hallman report some empirical evidence on the presence of a nonlinear cointegrating relationship between the US base money (M_0) divided by the consumer price index on the one side and the interest rate on three month US Treasury bills on the other side. They constructed an error correction model in which the deviation from the nonlinear attractor f had a significant coefficient. Granger and Hallman warn for a possible misinterpretation of the nonlinear attractor.

Since the values of the lower part of the attractor often also correspond to observations from the early part of the period of observation, it could well be that what is being interpreted as a nonlinear attractor could also be viewed as time varying linear cointegration between the variables [GH91a].

In [Sep94], Sephton has enlarged the setting; the ACE algorithm is replaced by the more powerful MARS algorithm [Fri91], and the case of more than two time series is considered. Sephton constructs critical values for the ADF test on MARS and gives some applications in which empirical evidence of nonlinear cointegration has been found, while the null hypothesis of no *linear* cointegration was not rejected.

Both MARS and ACE are general-purpose flexible regression algorithms, which do not provide a parametric representation of the nonlinear attractor. The problem with a nonlinear attractor found by some model free regression algorithm is that extrapolating the attractor's shape to situations not captured by the current data sample becomes very risky. In this case the interpretation of the nonlinear attractor as a long-run equilibrium for nonstationary time series seems no longer justified. Hence, future observations on nonstationary series typically are beyond the range of values encountered in the observed data sample. Having not assumed a parametric model for the cointegrating relationship, there is no reliable information on how to proceed in this novel area in the space of variables. This seems a natural consequence of estimating a cointegrating relationship by a flexible regression method.

The following example illustrates the reasoning employed in the above paragraph. Suppose

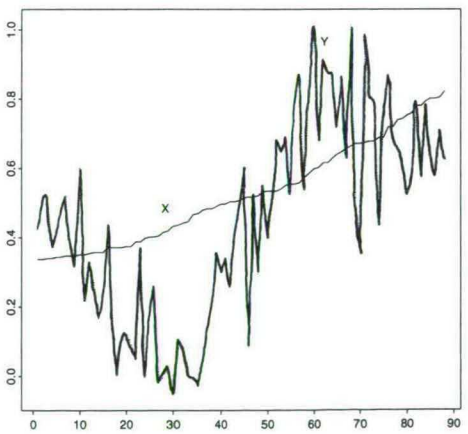

Figure 5.8: Two imaginative $I(1)$ economic time series

we have two imaginative economic time series x_t and y_t that are both $I(1)$ with time paths as depicted in Figure 5.8. There is no evidence that both series are linearly cointegrated. A neural network is then employed to investigate the possibility of nonlinear cointegration. Figure 5.9 shows the shape of the attractor found by the neural network. The null hypothesis

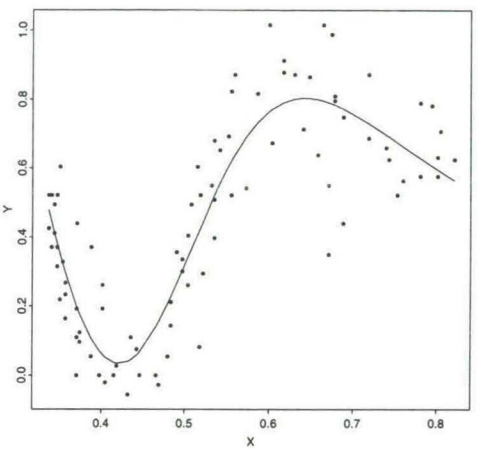

Figure 5.9: A nonlinear attractor between X and Y

of no (nonlinear) cointegration between x_t and y_t is rejected, which means that we have found empirical evidence for the existence of a *nonlinear* long-run relationship between x_t and y_t. Suppose an economist who predicts that X will reach the value of 1 within a year wants to know which Y-value the model would predict. A generally accepted way to proceed in the linear (parametric) case is to assume that the estimated relationship extends to the future. Once a linear attractor has been accepted as a satisfactory (correctly specified) long-run model and is correctly estimated, extrapolating it into the future is straightforward and uniquely determined. However, in the situation sketched in Figure 5.9, extrapolation is not so straightforward. Hence, the attractor is fitted by a flexible regression technique on the local information provided. The nonlinear attractor can be extrapolated in many ways, and no candidate extrapolation is justified or rejected by the set of observations. A parameterised neural network will provide an extrapolation, but its meaning is unclear. Hence, the neural network's parameters were estimated only to approximate the local training data appropriately; the values that the neural network provides for x-values outside the range of the training set are arbitrary. Although we have found evidence for the existence of a nonlinear attractor between y_t and x_t, the attractor constructed by a flexible regression method (such as the neural network) seems of little use when constructing

long-run expectations for Y, given projections of X. Nevertheless, the detection of nonlinear cointegrating relationships between economic variables may improve our understanding, and eventually the theories, of particular economic phenomena. We, therefore, develop a neural network test that helps to detect nonlinear cointegration.

5.4.3. Critical Values for ADF Tests on Neural Networks

To test for the existence of a nonlinear cointegrating relationship, we generalise the Dickey-Fuller tests, which have been presented in the previous section. The main issue is to construct critical values for cointegration tests on neural networks. In doing this, we adopt a similar procedure as employed in [EY87, Sep94], which was described in section 5.3. The cointegrating relationship is assumed to be represented by

$$x_{1t} = f(x_{2t}, x_{3t}, \ldots, x_{kt}) + \epsilon_t. \tag{5.15}$$

In (5.15) f is estimated by a neural network, but any other flexible regression method may be used instead.

Critical values of the ADF test on neural networks depend on several factors. Fitting neural networks according to the network construction procedure from Chapter 4, makes it necessary to condition critical values on the following neural network factors: number of hidden units, value of weight decay parameter, number of inputs, number of observations, and total number of restarts employed in finding the final network. All factors are somehow related to the tendency of neural networks to overfit the training data, which would result in unjustly small residuals. When the influence of the neural network factors is neglected, the ADF test would too often reject the null hypothesis of no cointegration.

The present soft- and hardware makes it computationally infeasible to construct tables of all possible combinations of neural network factors. In the applications, we will select the "best" neural network factors for a particular case, and will calculate the critical values that correspond to that particular combination of neural network factors.

The critical values are constructed under the null hypothesis of no cointegration through one thousand replications of the following procedure. Construct k independent random walks of length n, using the same data generating mechanism as in the previous section (5.3.3), that is,

$$\mathbf{x}_t = \mathbf{x}_{t-1} + U_t, u_{it} = 0.8\, u_{it-1} + \nu_{it}, \quad i = 1, \ldots, k, \tag{5.16}$$

with $\nu_{it} \sim$ i.i.d$(0, 1)$. Next, train a neural network with a particular set of factors to approximate f in (5.15). Then, estimate the parameters in

$$\Delta\hat{\epsilon}_t = \alpha_0 + \gamma_0 \hat{\epsilon}_{t-1} + \sum_{i=1}^{p} \gamma_i \, \Delta\hat{\epsilon}_{t-i} + \zeta_t, \quad \zeta_t \sim \text{i.i.d}(0, \sigma^2), \tag{5.17}$$

using the residuals from (5.15) and calculate the t-statistic of $\hat{\gamma}_0$ (usually minus signs are omitted). The one thousand t-statistics, so obtained, give an empirical distribution, which is used to calculate the critical values required for testing the null hypothesis of no cointegration.

Table 5.1: Some critical values for ADF test

factors					critical values		
n	k	N_h	λ	it	1%	5 %	10 %
100	4	2	0.0001	1	6.00	5.37	5.07
100	4	4	0.0001	1	6.83	6.13	5.73
100	4	2	0.1	1	4.03	3.48	3.15
200	4	2	0.0001	1	6.06	5.51	5.26
100	4	2	0.0001	5	6.67	5.95	5.60
100	4	0	0	1	4.61	4.02	3.71

Table 5.1 shows the critical values at different combinations of neural network factors.

The first five columns of the table show the particular combination of the neural network factors; n denotes the number of observations, k the number of variables (input+output units), N_h the number of hidden units, λ the weight decay parameter, and "it" the number of iterations employed to find good locally optimal weights. The next columns give the critical values for 1%, 5%, and 10% significance levels. These values behave as expected: the more flexible the neural network is, the larger the critical values become. The last row gives the critical values for the corresponding linear case, adopted from [EY87]. The critical values which correspond to the neural network generally exceed the critical values of the linear model, as expected. When, however, the decay parameter restricts the flexibility of the neural network heavily ($\lambda = 0.1$), the critical values are below the corresponding critical values of the linear model. In this case the neural network more or less acts as a restricted *linear* model.

5.4.4. An Example

A simulation experiment is designed to test the effectiveness of the neural network test for detecting nonlinear cointegration between $I(1)$ time series. Two random walk series x_{1t} and x_{2t} ($t = 1, \ldots, 200$) are constructed. Nonlinear cointegration between y_t, x_{1t}, and x_{2t} is enforced by

$$y_t = 0.5x_{1t}^2 + \log(x_{2t}^2) + \nu_t.$$

Figure 5.10 depicts the series.

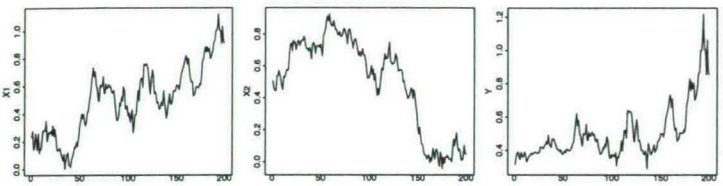

Figure 5.10: Three nonlinearly cointegrated $I(1)$ series

A test for linear cointegration by the Engle-Granger method (with four lags included in the ADF test of the residuals) gave an ADF-statistic of 2.47, which is below the 10% critical value of 3.47 [Mac91]. So, the null hypothesis of no (linear) cointegration is not rejected by the data.

A neural network with inputs x_{1t} and x_{2t}, two hidden units, a skip layer, and a weight decay parameter of 0.0001, is also fitted to the same data. The NN-residuals are tested for a unit root by the neural network ADF test with four lags included. The corresponding ADF-statistic is 6.14, and the simulated critical values are: 5.99 (1%), 5.35 (5%), and 5.09 (10%). So, the null hypothesis of no cointegration is (correctly) rejected by theneural network test at an error level of 1%.

We have repeated this experiment 100 times. The neural network version of the Engle-Granger test for cointegration rejected the null hypothesis of no cointegration 66 times at an error level of 10%. The standard Engle-Granger test rejected the null of no cointegration only 31 times at the same error level. The ideal test would reject 90 times.

This example shows that there are cases in which the standard Engle-Granger test fails to detect cointegration, but the neural network version of it does not fail. Therefore, the nonlinear generalisation of the cointegration concept does make sense.

5.4.5. Implications for the Short-run

In the foregoing sections we argued that it is risky to make long-run expectations of an $I(1)$ series based on a locally estimated nonlinear attractor. We think, however, that knowledge of the cointegrating relationship can still be exploited to model the short-run dynamics by an error correction model (ECM).

Granger and Teräsvirta [GT93] note that the theory of error-correction for nonlinear attractors is still incomplete. A difficulty is that the difference of an LMM series need not necessarily be SMM. To deal with this difficulty, Granger and Teräsvirta introduce the operator $\square_d$, which is defined by

$$\square_d x_t = x_t - \psi(x_{t-j}, j = 1, \ldots, d), \tag{5.18}$$

where d is the minimum integer such that the truncation ψ exists and $\square_d x_t$ is SMM. In practice,

ψ can be estimated by a nonparametric estimator (for instance, by a neural network). Note that the linear difference operator Δ_d is a special case of $\square_d$. We, however, have not been able to construct an example of an x_t that is LMM, does not 'explode'[2], and is nonlinear. Granger and Teräsvirta do not provide any example of such a process either.

Next, Granger and Teräsvirta [GT93] propose the following nonlinear form of the error-correction model:

$$\square_d y_t = \beta z_{t-1} + \sum_{i=1}^{l} \alpha_i \square_d x_{t-i} + \sum_{i=0}^{l} \alpha_i' \square_d y_{t-i} + \nu_t, \tag{5.19}$$

where $z_t = y_t - f(x_t)$, the deviation from the nonlinear attractor. They conclude that there is no theory or practical experience with these models.

If the difference of an LMM series results in an SMM series, which seems often to be case in practice, then we can proceed as in the linear case. Let $z_t = y_t - f(x_t)$ denote the deviation from the (nonlinear) equilibrium situation. The following ECM can then be formulated:

$$\Delta y_t = \alpha_0 + \beta z_{t-1} + \sum_{i=0}^{l} \alpha_i \Delta x_{t-i} + \sum_{i=1}^{l} \alpha_i' \Delta y_{t-i} + \epsilon_t. \tag{5.20}$$

In this equation the short-run part is assumed to be linear. In practice, however, it is the short-run dynamics that may well be nonlinear. Economic theory usually has not much to tell about the particular functional form of the short-run dynamics. Therefore, it seems promising to extend the ECM in (5.20) by a nonlinear part, parameterised by a neural network. Let $\mathbf{u}$ define $(\Delta x_t, \Delta x_{t-1}, \ldots, \Delta x_{t-l}, \Delta y_{t-1}, \ldots, \Delta y_{t-l})$. Then the neural network extension of (5.20) is

$$\Delta y_t = \alpha_0 + \beta_0 z_{t-1} + \alpha^T \mathbf{u} + \sum_{i=1}^{N_h} \beta_i \phi(\mathbf{w}_i^T \mathbf{u}) + \epsilon_t, \tag{5.21}$$

where N_h is the number of hidden units in the neural network. At time t variable Y will change, due to a disequilibrium situation and contemporaneous and lagged changes in X; in (5.21) the reaction to disequilibrium is assumed to proceed linearly. It is possible that Y reacts in a nonlinear way to the 'disequilibrium error' made in the previous period and onto lagged changes in X and Y. This can easily be implemented by a feed-forward neural network, including z_{t-1} in the $\mathbf{u}$-vector.

The nonlinear generalisation of the standard ECM will be used in Chapters 8 and 9. The ECMs are typically employed to construct short-term predictions; up to several periods ahead. So, probably extrapolation difficulties in the (eventually) nonlinear attractor are then of smaller concern.

[2]We mean: converges to infinity in a finite (small) number of steps.

Whether *ex ante* predictions would gain from our ability to detect nonlinear cointegration in the data sample still remains unclear, but it may help to enlarge our insight in past economic phenomena and in the sizes of specification errors that we make when restricted to linear models. The extension of the short-run part of the standard ECM by a feed-forward neural network seems to be practically more relevant.

5.5. Conclusions

The general aspects of economic modelling, which were introduced in Chapter 1, hold for both cross-sectional and time series data. The latter, however, can cause additional difficulties, in particular when the time series is nonstationary. To distinguish between stationary and nonstationary time series, Dickey-Fuller and augmented Dickey-Fuller tests, which test for the presence of unit-roots in an observed time series, were introduced. When these tests confirm the hypothesis of integratedness of an observed time series, then standard statistical inference is no longer valid; cointegration analysis should be performed instead.

Cointegration analysis helps in finding 'true' long-run relationships among nonstationary economic series instead of spurious ones. When evidence for a long-run relationship has been found, the next step is often the formulation of an error-correction model (ECM). ECMs combine short-run effects (based on time series theory) and long-run effects (based on economic theory) into a single model. The main feature of an ECM is its use of the discrepancy from the long-run model in the previous period as as an explanatory variable in the short-run model.

The definitions of cointegration and error-correction implicitly assume linearity. When using flexible regression models, nonlinear generalisations of these concepts comes to mind. We noticed that there is not much theory on nonlinear cointegration and nonlinear error-correction modelling. Neural networks may help in making these nonlinear generalisations operational. The neural network ADF test, which we developed, properly indicated the presence of nonlinear cointegration in an artificial example. The extrapolation of a nonlinear cointegrating relationship constructed by a neural network, causes conceptual difficulties. Economic understanding, however, may be improved by a confirmed nonlinear cointegrating relationship for a data sample at hand.

Neural networks seem particularly suited for implementing nonlinear error-correction models. The reasons are twofold. First, it is unclear why short-run dynamics are best modelled linearly. Second, economic theory has usually not much to say about the particular parametric form of the short-run dynamics to employ. The nonlinear extension of the standard ECM is performed simply by the addition of an additional term, which represents a feed-forward neural network. All parameters in the nonlinear ECM are determined by a regular neural network learning algorithm.

The issues described in this chapter and in the previous chapters will be employed in the case studies of part II. An aim of these case studies is to learn about the relevance of these new concepts in economic modelling.

Part II

Applications

Chapter 6

Neural Network Applications to Economics and Finance: An Overview

6.1. Introduction

In the previous chapters neural networks were discussed from statistical and methodological perspectives. This chapter forms a transition from these methodological aspects of neural networks to the application of neural networks to economic and financial case studies in the next three chapters. Various financial and economic problems to which neural networks have been applied are reviewed. We have no intention to provide a complete overview; we only sketch popular application areas and indicate some difficulties in the studies that have been performed.

During the last few years there has been a growing interest in applying NNs to problems in the domain of economics and finance. Numerous conferences have addressed this topic, such as "Neural Networks in the Capital Markets (NNCM)", "International Workshop on Parallel Applications in Statistics and Economics (PASE)", and "International Workshop on Artificial Intelligence in Economics and Management (AIEM)". In 1994, at least four journals had special issues devoted to neural networks: The International Journal of Forecasting, Vol. 10; Econometric Reviews, Vol. 13 (No. 1); Decision Support Systems, Vol. 11; and Simulation Vol. 62 (No. 5).

Many articles have been written overviewing the potential of neural networks for finance; among them are [TG91, Hop93, HJR90]. These types of contributions are particularly useful for financial managers who want to get a quick impression of what neural networks are, which types of financial problems they are suited for, and how they relate to other AI or statistical techniques.

Trippi and Turban [TT93] with their collection of 28 articles provide a valuable source of information on neural networks applications in economics and finance. They subdivided the

collection of articles into five parts: analysis of companies' financial condition, business failure prediction, debt risk assessment, security market applications, and financial forecasting. In most cases the financial problem is represented as a classification problem.

Baestaens *et al.* [BvdBW94] provide another source of information on neural networks applied to trading in financial markets. They examined the usefulness and performance of neural networks for the following financial topics: crashes and panics in financial markets, predicting cash flows (tax receipts), European option pricing, stock market indices, international portfolio management, credit risk assessment, corporate failure prediction, and technical trading.

Weigend and Gershenfield [WG94] provide a valuable source of information on neural networks for time series prediction. This work is the result of the 1992 Sante Fe time series competition. The purpose of the competition was to compare the prediction performance of different time series analysis techniques on the same data sets. The participants had a choice of six data sets. Among the data sets there was one set in the financial domain: the prediction of high-frequency currency exchange rate data.

This chapter overviews a -necessarily- small sample of the vast amount of literature on neural network applications in economics and finance. We will categorise the literature, using the statistical problem type that is addressed, discerning four categories: econometrics, multivariate regression, classification, and (univariate) time series analysis.

The outline of the chapter is as follows. In section 2 we review the literature on neural networks in econometric testing. Section 3 reviews neural networks for multiple regression and classification problems in economics and finance. In section 4 we review the literature on neural networks in time series prediction. Section 5 concludes the chapter.

6.2. Econometrics

In the econometric area, much theoretical work on neural network learning has been done by White and his co-workers (see [Whi92, KW92]). White, as an econometrician, is mainly concerned with the development of theories on estimation and inference for neural networks that are comparable to existing theories for nonlinear dynamic models. White's contributions ([Whi92, KW92]) do provide a solid and rigorous basis for an asymptotic distribution theory for the optimal network weights, which may help in statistical generalisability, such as asymptotic confidence intervals, prediction intervals, and tests of hypotheses [MKA94].

White [Whi89a] used the general feature extraction capabilities of neural networks to develop a new statistical test for neglected nonlinearities in linear models. The neural network test is of the Lagrange multiplier type, which generally tests for certain restrictions on the parameters in a parametric model, while only estimating the restricted model (see, for instance, [Tho93b]).

The test uses a neural network with a single hidden layer (of size q) augmented by direct

connections from input to output; in Chapter 3 we saw that for such a network the output $\hat{y}$ is calculated as

$$\hat{y} = \theta^T \mathbf{x} + \sum_{i=1}^{q} \beta_i \, \phi(\gamma_i^T \mathbf{x}). \tag{6.1}$$

When the null hypothesis of linearity is true, the optimal network weights β_i, say β_i^*, are zero ($i = 1, \ldots, q$). The capability of $\sum_{i=1}^{q} \beta_i^* \, \phi(\gamma_i^T \mathbf{x})$ to extract structure from $e_t^* = y_t - (\theta^*)^T \mathbf{x}$ will give power to the neural network test.

An obstacle in the straightforward application of the usual tools of statistical inference is that under the null hypothesis the network weights γ_i are not identified. Different ways can be followed to resolve this difficulty. The simplest procedure, which is employed in [Whi89a, LWG93], is to randomly select the γ_i-parameters. Kuan and White [KW92] consider the alternative of optimising the direction in which nonlinearity is sought by choosing the γ_i-values.

The neural network test with randomly selected γ_i-values has good power [LWG93, GT93]: "It thus appears to be a useful addition to the modern arsenal of specification testing procedures" [KW92].

The work of White definitely makes neural networks accessible to econometricians [KW92], although for many neural network engineers most of White's theories are inaccessible due to the high mathematical level employed in his articles.

Some people disagree about the practical value of these asymptotic statistical theories. For instance, Maasoumi *et al.* [MKA94] state that the asymptotic statistical aspects should at least not be overemphasised. They further believe that one should *not* eschew analysis of systems and data with neural networks unless one can draw statistical inferences. This belief is confirmed by the literature on neural networks; most of the articles we are familiar with refrained from making statistical inferences.

6.3. Multiple Regression and Classification

Multiple regression problems combine information on several selected variables to approximate the variable of interest as close as possible. The variable of interest, for instance stock-price, typically is real-valued. Classification problems are conceptually very close to the classical regression problem with cross-sectional data. The only difference is that the variable of interest is a class label, not a continuous or discrete variable. In most applications there are only two classes (e.g., bankrupt firms and surviving firms), although problems with more than two classes also occur (e.g., in the bondrating problem firms can be rated into one of nine categories).

Classification problems are approached by standard regression techniques or by special purpose (statistical) techniques, such as logistic regression or linear discriminant analysis (lda). In

empirical studies the performances of neural networks are often contrasted with the performances of either multiple regression, logistic regression, or linear discriminant analysis.

The explanatory variables in classical regression and classification are of the cross-sectional data type; in economics and finance, however, the variables in a regression can be time series as well. In practice regression and classification models are constructed for the purpose of forecasting, explanation, or decision making. A decision model tries to predict a human's judgement, based on the factors that a person would use in making the judgement. In contrast with expert systems, which model human decision making by explicit rules, neural networks acquire knowledge automatically by learning from historical cases rather than by intensive expert interviewing, which takes much time and money.

Examples of regression problems in economics and finance are: stock price prediction [Sch90, RP91, GO93], futures price prediction [GO93], stock performance modelling [RZF94], risk management in mortgage-backed securities portfolio management [BKW93], and predicting trading volume on the New York Stock Exchange [WL94].

Marquez *et al.* [MHWR94] performed a comparative simulation study to assess the potential of neural networks as an alternative to classical (parametric) regression. The general approach was to generate data representing common functional forms (linear, logarithmic, and reciprocal) encountered in regression modelling; next neural networks are compared to regression models using that data. Hundred sets of n points (15, 30, or 60), each with three noise levels ($R^2 = 0.3$, 0.6, and 0.9), were created for each functional form. The fit of the neural network was compared with the fit of the true regression on hold-out data sets of 100 points. Marquez *et al.* [MHWR94] found the overall MAPEs (mean average prediction error) for the neural network to be very good; less than 2% away of the error rate of the true functional form. They conclude that the NN's ability to work well when the functional form is unknown, makes neural networks an attractive choice in many applications.

Although neural networks show promising results in regression tasks using simulated data, in practice there may exist difficulties in measuring a particular entity, such as, e.g., consumer confidence, which makes any form of modelling doomed to fail –neural networks included.

Gorr *et al.* [GNS94] provide an example of a human judgement problem, namely, graduate school admission decisions. They performed a cross-section study in which a multivariate regression model is used to predict student grade point averages (GPA). The paper compares the neural network model with linear regression, stepwise polynomial regression, and an index used by the admission committee for predicting GPAs. Although the neural network identified additional model structure over the regression models, none of the empirical methods was significantly better than the practitioners' index, according to the statistical tests used by the authors. The authors suggest that this result may be due to difficulties in measuring student motivation and perseverance, and the lack of discriminative power of some predictors. Such

difficulties often arise in economic modelling.

The rating of bonds is another example in which human judgement plays an important role. Several researchers [DS94, MU92, MU94, SS94, KWR93, DKV95] have used neural networks to predict the rating of corporate bonds on the basis of a set of financial indicators calculated from the balance sheets of those firms. In the literature the bondrating problem has been represented both as a regression and as a classification problem, depending on how many different ratings are discerned.

Dutta and Shekhar [DS94] used ten explanatory factors, a training sample of 30 firms, and a test sample of 17 firms. They restricted the bond rating problem –probably forced by the small sample size– to predicting whether a bond is rated as AA or non-AA. Drawn on this very limited experiment, Dutta and Shekhar conclude that neural networks consistently outperformed linear regression. Surkan and Singleton [SS94] divided 56 bond rating exemplars into a training set of size 16 and a test set of size 40. Seven factors were used to classify the bond ratings into two classes: Aaa (highest quality) and A1, A2, A3 (investment grade, but lower quality). They found a neural network with two hidden layers, with 10 and 5 hidden units respectively, to perform better than linear discriminant analysis on the test set.

The above studies suffer from several technical and methodological problems; in particular, the number of data sets is limited, the data sets are small, and the neural network construction procedure is not well described. A better study on the bond rating problem is Moody and Utans [MU94]. They used a data set of 196 firms, 10 financial ratios reflecting the fundamental characteristics of the firms, and 5-fold cross-validation to estimate the prediction error of the neural network. In contrast with previous studies, Moody and Utans distinguish 18 different ratings. They describe their neural network construction procedure very well, which makes it a valuable contribution. Moody and Utans found a linear regression model predicting 80.5% of the data within two notches from the correct target; the best neural network architecture, which used only two inputs, predicted 87.5% of the data within two notches.

Besides the bondrating problem, there are numerous other examples of classification problems in economics and finance, for example: bank failure prediction [TT93, Part 3][TK92, FG93], credit scoring [TT93, Chapter 16], and market response modelling [DDG94].

Tam and Kiang [TK92] performed a well conducted comparative study, including neural networks, lda, logistic regression, ID3, and k-nearest neighbour classification, to the prediction of Texas bank failures in the period 1985-1987. Their data sample consisted of 118 bank data (59 nonfailed and 59 failed –one year and two years prior to failure) for training, and 44 bank data (22 failed and 22 nonfailed) for testing. Each bank is described by 19 financial ratios. Tam and Kiang modified the standard least squares error criterion used in the back-propagation algorithm to include prior probabilities of each group and their misclassification costs. Besides ranking the methods by misclassification rates on the test set for different values of misclassification costs

and prior probabilities, they also estimated the misclassification rates by the jackknife method and ranked the classifiers accordingly. Their empirical experiments show that neural networks give better predictive accuracy than lda, logistic regression, k-nearest neighbours, and ID3.

Dasgupta *et al.* [DDG94] compare two statistical market response models (logistic regression and lda) to a neural network model. The goal of modelling is to identify consumer segments based upon their willingness to take financial risks and to purchase a non-traditional investment product. The empirical analysis is conducted using two cross-sectional survey data sets (on an individual level) related to the market of financial services. The two data sets are subdivided into a training set and a test set as follows: (531; 183) and (616; 213). If the performance of the three models (measured as percentage correctly classified) are rank ordered, the neural network model performs better than the other two models, for both data sets. The improvement, however, is only marginal; statistical testing revealed that at reasonable significance levels the null hypothesis of no difference in predictive accuracy could not be rejected.

6.4. Time Series Prediction

In time series prediction, future values of the variable of interest are predicted directly from the series' own history. Studies in which neural networks are applied to such problems, almost always use a large set of high frequency data (daily or weekly). Three practical advantages of a time series approach opposed to, for instance, a regression approach are: collecting data is easy, economic theory is not needed, and *ex ante* predictions are straightforwardly constructed.

In several studies neural networks have been compared to traditional time series forecasting techniques. The best of these use a sample from the well known 'M-competition' [MCF+82], in which 1001 real time series were gathered. In [SP94, TAF91] the forecasting ability of neural networks was compared to that of the Box-Jenkins time series technique. Both studies found that for long-memory time series the two methods perform equally well, but for short-memory time series the neural network outperformed the Box-Jenkins approach.

Hill *et al.* [HMOR94] review other studies that assess neural networks for time series forecasting, among them is their own study [HOR94]. In the latter study, they compared time series models across yearly, quarterly, and monthly data from a systematic sample of 111 M-competition time series. They found that neural networks were significantly better than statistical and human judgement methods by about 5% MAPE (mean absolute prediction error) in the quarterly time series and about 2% MAPE in the monthly time series. Hill *et al.* made two observations. First, neural networks showed better performance at predicting monthly series than at predicting quarterly or yearly series. Second, the superiority of neural networks was in the later periods of the forecast horizon -which is confirmed in [TAF91]. They attribute these findings to the presence of nonlinear patterns in the data, which are advantageous for neural

networks. Aggregated data, such as yearly time series, tend to have fewer nonlinearities than monthly series. So, the opportunities for neural networks are larger with monthly time series than with yearly time series. If the time series contains nonlinear patterns, the deviation of linear models from the target series will increase with the prediction horizon. If the neural network, on the other hand, has detected the nonlinear pattern, its improvement in long-run prediction accuracy will be increasingly apparent [HMOR94].

Two popular financial time series prediction problems to which neural networks are regularly applied are: currency exchange rate ([WG94, page 219-263][RABCK93]), and stock price prediction [Whi88, Sch90].

White [Whi88] performed one of the early studies on the usefulness of NNs for stock price prediction. White's results turned out to be disappointing; no evidence was found against the simple efficient market hypothesis. This paper is one of the few empirical papers that reported disappointing results. In his conclusion White states "the present neural network is not a money machine"; financial traders relying on neural networks hope he is wrong.

Schoeneburg [Sch90] analysed the possibility of predicting stock prices of the German stocks BASF, COMMERZBANK, and MERCEDES, on a short-term, day-to-day basis with the help of neural networks. His results made the author expect that in the future NNs could considerably improve the prognosis of stock prices. Despite these encouraging results, some problems with respect to neural network architecture design were recognised. Other studies (e.g. [Col91, RABCK93, MJ94, PDM+92]) go even beyond merely predicting stock prices: they want a neural network to learn and, consequently, generate buy-sell decisions in a trading system. The imaginary profits were reported to be high; see [Col91, RABCK93, MJ94, PDM+92].

The Santa Fe time series competition [WG94] offered a financial data set consisting of quotes, on a time scale of one to two minutes, for the exchange rate between the Swiss franc and the U.S. dollar. The exchange rate market is based on bids and asks to buy and sell. The complete data range from May 20, 1985 to April 12, 1991 –which corresponds to 11.5 MB (Mega Bytes). The organizers held back the data from the period August 7, 1990 to April 18, 1991 to evaluate the submission of the competitors. The quality of the predictions is expressed in terms of the following ratio of squared errors:

$$\frac{\sum_t(\text{observation}_t - \text{prediction}_t)^2}{\sum_t(\text{observation}_t - \text{observation}_{t-1})^2}.$$

The denominator represents the prediction error made by the random walk. A ratio above 1.0 thus corresponds to a prediction that is worse than chance; a ratio below 1.0 is an improvement over the random walk model. [WG94] stated that some of the submitted predictors were worse than chance by a factor of 16! Table 6.1 gives the out-of-sample prediction results of the two best submissions, which are due to Mozer [WG94, pp.243-264] and Zhang & Hutchinson [WG94, pp.219-241]. These forecast are made for 1 minute, 15 minutes, and 60 minutes after the last

Table 6.1: Prediction performance on the financial data set of the best competitors of the Santa Fe time series competition

	1 minute	15 minutes	60 minutes
Mozer	0.9976	0.9989	0.9965
Zhang & Hutchinson	1.090	1.103	1.098

tick. The results confirm the efficient market hypothesis: the best prediction of tomorrow's rate is today's rate. Hence, the neural network was not able to provide better predictions for the exchange rate than the random walk model.

Neural networks are also used as an alternative to specialised nonlinear models from finance. Donaldson *et al.* [DKK93] model the conditional volatility in stock returns of stock index data for the Tokyo, London, New York, and Toronto exchanges, by several popular volatility models, such as members of the ARCH (autoregressive conditioned heteroscedasticity) family, flexible Fourier functions (FFF), and neural networks. They applied numerous statistical performance tests to the various models for different stock indices. Based on these tests, they conclude that if the information set is constrained to past returns only, a flexible form such as the neural network model may out-perform fully parametric methods such as the ARCH family.

6.5. Conclusions

The selected set of articles almost uniformly assigns good prediction power to neural networks. Neural networks performed as well or better than alternative statistical models. However, many of these studies suffer from technical and methodological problems. First, small sample sizes make reliable prediction error calculation troublesome. In particular, estimation of the prediction accuracy on a single hold-out set, which is almost common practice, may cause large variability in the error estimates (see [WL94]). Second, the comparison of predictors is typically restricted to a mere ranking of some error criterion; [TK92, DKK93] form positive exceptions. Third, the procedure of neural network construction is often insufficiently documented or is badly performed; for example, authors take no steps to guard against overfitting, or neglect the occurrence of multiple local minima. Fourth, we conjecture that the levels of expertise of the researchers in the techniques incorporated in a comparative study generally vary significantly among the various techniques. This may obscure the results. Competitions such as the Santa Fe time series study [WG94] and the StatLog project [MST94] avoid this difficulty, and are consequently of great importance to achieve a fair comparison of the various competing techniques. Fifth, in case studies it is generally difficult to explain the dominance or failure of a particular

method. Additionally, theoretical or experimental simulation studies are needed to examine the differences among the competing modelling techniques.

In the previous chapters we extensively described the economic modelling process and the role neural networks may play, the neural network construction procedure, and the evaluation of neural network models. Additionally, we found very few articles that examine the practical use of neural networks for the econometric modelling of time series.

In the subsequent chapters we will apply neural networks to three economic case studies, namely, the prediction of hedonic house prices, the prediction of the production of new mortgage loans, and the prediction of exchange rates.

Chapter 7

Modelling the Hedonic Price for Housing in Boston

7.1. Introduction

This chapter examines the potential of neural networks in a modelling case with cross-sectional data, namely the construction of a hedonic model for house prices. The problem is adopted from a paper by Harrison and Rubinfeld [HR78], in which a "hedonic" price index for housing is estimated for use in a subsequent estimation of the marginal willingness-to-pay for clean air. *The basic principle of the hedonic approach to economics is that each consumer good is regarded as a bundle of characteristics for which an implicit valuation exists [Jan92].* This principle allows us to regard a good's *price* in the same way. Harrison and Rubinfeld regard each house as a bundle of characteristics (among others the level of air pollution), and the price of each house as reflecting the value of its characteristics. Let H^p denote the house price and x_i object characteristic i. Then the house price equation may be written as:

$$H^p = g(x_1, \ldots, x_q), \tag{7.1}$$

where q denotes the number of object characteristics. Janssen [Jan92] provides an in-depth discussion of hedonic models and their applications. He constructed hedonic house price models for four cities in the Netherlands.

Hedonic house price models, when estimated sufficiently accurate, can be utilised in the automatic appraisal of house values. Local authorities require house values in order to calculate the amount of property tax due. Automating the appraisal process will reduce its costs and will increase its effectiveness [Jan92].

For this particular problem, there is no theoretical knowledge that proposes a specific functional form for the relationship g in (7.1). Therefore, it looks promising to employ a data

driven approach to model specification. We apply the neural network methodology, as outlined in Chapter 4, to this modelling problem.

The outline of the chapter is as follows. In section 2 the modelling performance of the neural network is contrasted with the modelling performance of two linear models; the first is linear in both parameters and variables, and the second is linear in the parameters, but also includes nonlinearly transformed explanatory variables as used in [HR78, BKW80]. In section 3 the out-of-sample prediction accuracy achieved by the neural network is statistically compared to the prediction accuracies achieved by the two linear models. When a neural network has been found that fits the data well, we want to extract information on the (complex) relationship. In section 4 we propose some aids that support the investigator in 'understanding' the modelled relationship. Section 5 concludes the chapter.

7.2. The Modelling Process

In the original paper due to Harrison and Rubinfeld [HR78] the attributes shown in Table 7.1 were used to valuate the price for houses. The original paper focussed on the impact of air pollution on the prices for houses. The variable of interest, the house price H^p in (7.1), is denoted by MEDV. The data are of the cross-sectional type, i.e., the attributes are measured across various suburbs of Boston, at a particular point in time.

7.2.1. The Data

The data set consists of 506 instances and was taken from the StatLib library maintained at Carnegie Mellon University. The basic data, which are also listed in [BKW80], are a sample of census tracts in the Boston Standard Metropolitan Statistical Area in 1970.

Figure 7.1 and Table 7.2 reveal some characteristics of the data: the distributions of the values of each attribute are shown in Figure 7.1; the matrix of cross-correlations between all attributes is presented in Table 7.2. Useful information can be extracted at a glance. The last row in the correlation matrix suggests, for example, that the number of rooms per dwelling (RM) and the % lower status of the population (LSTAT) are important determinants of the housing value. The direction of influence corresponds with common sense: more rooms will in general result in a higher housing value, and a high percentage of lower status of the population will decrease the value of a house. Another example is the correlation between NOX and INDUS (0.8), which says that industrial areas are more polluted than rural areas.

Table 7.1: Definition of variables in (7.1)

symbol	definition
CRIM	per capita crime rate by town
ZN	proportion of residential land zoned for lots over 25,000 sq.ft.
INDUS	proportion of non-retail business acres per town
CHAS	Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
NOX	nitric oxides concentration (parts per 10 million)
RM	average number of rooms per dwelling
AGE	proportion of owner-occupied units built prior to 1940
DIS	weighted distances to five Boston employment centres
RAD	index of accessibility to radial highways
TAX	full-value property-tax rate per $10,000
PTRATIO	pupil-teacher ratio by town
B	$1000(Bk - 0.63)^2$ where Bk is the proportion of blacks by town
LSTAT	% lower status of the population
MEDV	Median value of owner-occupied homes in $1000's

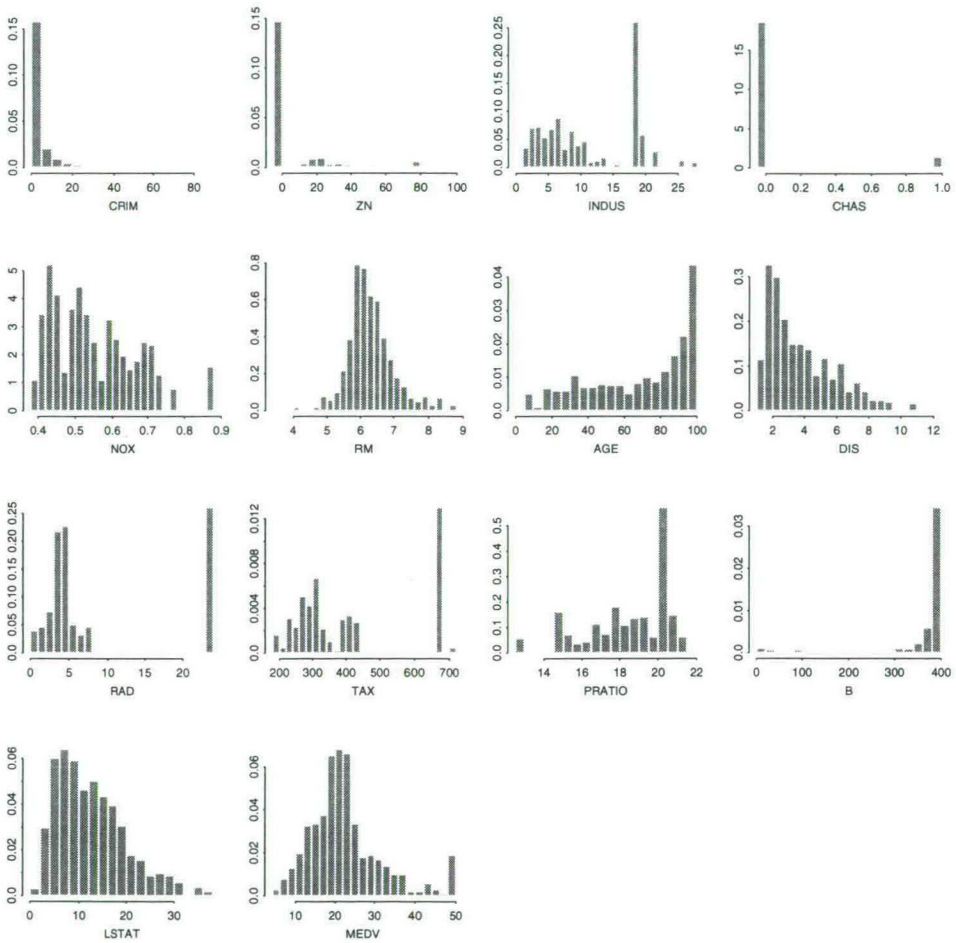

Figure 7.1: The distribution of attribute values.

Table 7.2: All pairwise cross-correlations

	CR	ZN	IND	CH	NO	RM	AG	DIS	RAD	TAX	PT	B	LS
CRIM	1.0	-0.2	0.4	-0.1	0.4	-0.2	0.4	-0.4	0.6	0.6	0.3	-0.4	0.5
ZN	-0.2	1.0	-0.5	-0.0	-0.5	0.3	-0.6	0.7	-0.3	-0.3	-0.4	0.2	-0.4
INDUS	0.4	-0.5	1.0	0.1	0.8	-0.4	0.6	-0.7	0.6	0.7	0.4	-0.4	0.6
CHAS	-0.1	-0.0	0.1	1.0	0.1	0.1	0.1	-0.1	-0.0	-0.0	-0.1	0.0	-0.1
NOX	0.4	-0.5	0.8	0.1	1.0	-0.3	0.7	-0.8	0.6	0.7	0.2	-0.4	0.6
RM	-0.2	0.3	-0.4	0.1	-0.3	1.0	-0.2	0.2	-0.2	-0.3	-0.4	0.1	-0.6
AGE	0.4	-0.6	0.6	0.1	0.7	-0.2	1.0	-0.7	0.5	0.5	0.3	-0.3	0.6
DIS	-0.4	0.7	-0.7	-0.1	-0.8	0.2	-0.7	1.0	-0.5	-0.5	-0.2	0.3	-0.5
RAD	0.6	-0.3	0.6	-0.0	0.6	-0.2	0.5	-0.5	1.0	0.9	0.5	-0.4	0.5
TAX	0.6	-0.3	0.7	-0.0	0.7	-0.3	0.5	-0.5	0.9	1.0	0.5	-0.4	0.5
PTRATIO	0.3	-0.4	0.4	-0.1	0.2	-0.4	0.3	-0.2	0.5	0.5	1.0	-0.2	0.4
B	-0.4	0.2	-0.4	0.0	-0.4	0.1	-0.3	0.3	-0.4	-0.4	-0.2	1.0	-0.4
LSTAT	0.5	-0.4	0.6	-0.1	0.6	-0.6	0.6	-0.5	0.5	0.5	0.4	-0.4	1.0
MEDV	-0.4	0.4	-0.5	0.2	-0.4	0.7	-0.4	0.2	-0.4	-0.5	-0.5	0.3	-0.7

7.2.2. Linear models

As said before, for the Boston house price problem, there is no theoretical knowledge that prescribes a specific functional form of the relationship between MEDV and the other attributes. An obvious start to specify the model, which is also made in [Jan92], is to fit a linear model (in both parameters and variables) to the data:

$$\begin{aligned} MEDV &= \alpha_0 + \alpha_1 CRIM + \alpha_2 ZN + \alpha_3 INDUS + \alpha_4 CHAS + \alpha_5 NOX \\ &\quad + \alpha_6 RM + \alpha_7 AGE + \alpha_8 DIS + \alpha_9 RAD + \alpha_{10} TAX \\ &\quad + \alpha_{11} PTRATIO + \alpha_{12} B + \alpha_{13} LSTAT + \epsilon. \end{aligned} \tag{7.2}$$

The results are shown in Table 7.3; AGE and INDUS are not significant (at a 5% level) on MEDV, so they are left out.

Although the signs of the estimated coefficients correspond to what is expected from economic or common sense knowledge, graphical inspection of plots of the residuals against each attribute and against estimated MEDV provides evidence of a misspecified functional form. The usual strategy is to transform the variables which seem to affect the dependent variable nonlinearly by some parametric function (e.g., $\log x$, $1/x$, or x^2), as suggested by the various plots. In this way, it can be quite time consuming to find the right functional form; the investigator has to search manually for a suitable functional form, using the data at hand.

In [HR78] the following model linear in the parameters is proposed and examined for fit:

$$\log(MEDV) = \alpha_0 + \alpha_1 CRIM + \alpha_2 ZN + \alpha_3 INDUS + \alpha_4 CHAS + \alpha_5 NOX^2$$

$$+\alpha_6 RM^2 + \alpha_7 AGE + \alpha_8 \log(DIS) + \alpha_9 \log(RAD) + \alpha_{10} TAX$$
$$+\alpha_{11} PTRATIO + \alpha_{12} B + \alpha_{13} \log(LSTAT) + \epsilon. \quad (7.3)$$

So, in contrast with model (7.2), this model includes several log and square-transformations of the variables. The effect of these transformations is an increase in R^2 to 0.81 (measured in back-transformed values), so a better fit is indeed obtained. The estimated coefficients, the standard errors, and the corresponding t-values of model (7.3) are presented in Table 7.4. A neural network offers an alternative to this manual transformation approach. It should be able to make an approximation to the data automatically that at least is as good as (7.3). The next section investigates whether this is possible.

Table 7.3: The OLS estimates (with standard errors) of (7.2).

attribute	value	st. error	t-value
(Intercept)	36.5	5.10	7.14
CRIM	-0.11	0.033	-3.29
ZN	0.046	0.014	3.38
CHAS	2.69	0.86	3.12
NOX	-17.77	3.82	-4.65
RM	3.81	0.42	9.12
DIS	-1.48	0.20	-7.40
RAD	0.31	0.066	4.61
TAX	-0.012	0.0038	-3.28
PTRATIO	-0.95	0.13	-7.28
B	0.0093	0.0027	3.47
LSTAT	-0.52	0.051	-10.35
R^2	0.74		

7.2.3. A neural network model

The previous section indicated that nonlinearities are present in the house price equation. In this section a neural network is used to explore possible nonlinearities. Neural network models are built according to the strategy described in Chapter 4.

All attributes (except MEDV) are used as inputs to the neural network with skip-layer connections. Network weights are determined by minimising the standard squared error loss

Table 7.4: The OLS estimates (with standard errors) of (7.3).

attribute	value	st. error	t-value
(Intercept)	9.76	0.15	65.22
CRIM	-0.012	0.0012	-9.53
CHAS	0.091	0.033	2.75
NOX^2	-0.0064	0.0011	-5.64
RM^2	0.0063	0.0013	4.82
log(DIS)	-0.19	0.033	-5.73
log(RAD)	0.096	0.019	5.00
TAX	-0.00042	0.00012	-3.43
PTRATIO	-0.031	0.0050	-6.21
B	0.36	0.10	3.53
log(LSTAT)	-0.37	0.025	-14.84
R^2	0.81		

function (also used by the regression models) plus the sum of squared weights penalty term, which has been called weight decay in section 3.4. The selection of the number of hidden units and the value of the weight decay parameter is based on 10-fold cross-validation. Neural network training and parameter selection is done on 80% of the data (randomly drawn); the remaining 20% is reserved for model evaluation. Ten multiple restarts with different randomly selected initial weight vectors are performed to "ensure" a good locally optimal network solution.

Table 7.5: NN selection.

N_h	weight decay value λ					
	0.5	0.1	0.05	0.01	0.001	0
0	0.72/0.70	0.73/0.70	0.73/0.70	0.73/0.70	0.73/0.70	0.73/0.70
2	0.72/0.70	0.78/0.70	0.82/0.77	0.88/0.79	0.89/0.78	0.90/0.78
4	0.73/0.70	0.73/0.70	0.85/0.73	0.91/0.81	0.93/0.81	0.95/0.61
6	0.73/0.69	0.78/0.69	0.85/0.78	0.91/0.84	0.95/0.84	0.96/0.57

note: cells display R^2_{in}/R^2_{cv}

corresponding values for model (7.3) are: 0.80/0.78

The intermediate results of the model selection process are displayed in Table 7.5. The cells display the in-sample and out-of-sample coefficient of determination (R^2) for each neural

network characterised by the network parameters N_h (number of hidden units) and λ (weight decay value). The in-sample R^2, which is denoted by R^2_{in}, represents the R^2 of the final neural network when fitted to 80% of the data. The out-of-sample R^2, denoted by R^2_{cv}, is calculated from the vector of predictions obtained during the cross-validation procedure (also on the same 80% of the data).

The neural network model with the highest R^2_{cv} is selected as the final neural network model for prediction purposes. Table 7.5 indicates that the best network consists of 6 hidden units and employs a weight decay value of 0.01 during weight estimation. According to the R^2_{cv} criterion this neural network model improves over the parametric model found after manually transforming some of the variables (7.3): from 0.78 to 0.84.

The remaining 20% of the data (106 observations), which were randomly selected from the total sample, are used to assess the out-of-sample prediction accuracy of the final network. In Table 7.6 the out-of-sample and in-sample R^2 of the neural network are compared to the out-of-sample and in-sample R^2 of the parametric models (7.2) and (7.3). The neural network model automatically finds an approximation that is clearly better than the simple linear model and even the model used in [BKW80]; both in-sample and out-of-sample.

Table 7.6: In-sample and out-of-sample R^2 of the neural network model, the simple linear model (7.2), and the transformed linear model (7.3).

model	in-sample R^2	out-of-sample R^2
linear model (7.2)	0.73	0.77
transf. linear model (7.3)	0.80	0.86
neural network model	0.91	0.90

7.3. Model comparisons

In the previous section the performances of the different models were compared. Implictly or explicitly a study often intends to select a "winner".

We statistically compare the performance of the final neural network to the performances of the models (7.2) and (7.3), on the same randomly chosen hold-out set. Remember that the data set was split into two parts: the first part (400) to estimate the parameters of the model; the second (106) part to measure the model's performance. Let PE_k denote the vector of squared prediction errors of model k

$$\mathrm{PE}_k = \{(H_i^p - \hat{H}_i^p)^2\}_{i=1}^{106}. \tag{7.4}$$

Pairwise t-tests are performed to test for the differences between the average prediction error of the neural network model NN and model (7.2), and between the neural network model and model (7.3). A plot of the quantiles of the distribution of the PE_k-values against the quantiles of a normal distribution reveals that the PE_k-values are not normal (Gaussian) distributed (fat tails). We, therefore, employ a t-test based on bootstrap resampling (see section 3.5). The p-values, which are based on 5000 resamples, are displayed in Table 7.7. Table 7.7 shows the

Table 7.7: Pairwise t-tests

difference	p-value	adj. p-value
NN-model (7.2)	0.000	0.000
NN-model (7.3)	0.0244	0.0482

raw p-values of the two pairwise tests and the Šidàk adjusted p-values, which take account of the multiplicity effect. Due to the limited number of tests performed, the multiplicity effect plays a minor role. So, we took the simple approach towards adjustment of the raw p-values. When many tests are performed, it is recommended to use the more elaborate multiple comparison tests described in section 3.5 to avoid too conservative conclusions.

So, rather than providing a mere ranking, hypothesis testing shows whether a significant difference between functions is found. From the statistical analysis (Table 7.7) we conclude that the neural network provides a significantly higher accuracy in predicting house prices, when compared with the simple linear model (7.2) and with the transformed model (7.3).

7.4. Analysis of the final network

A frequently heard disadvantage of neural networks is difficult interpretation of the approximating function. However, if complex relationships characterise the data, it seems unreasonable to expect the approximation to be easily interpretable. If, on the other hand, relatively simple nonlinearities are present in the data (e.g. polynomial terms), the network solution may conceal this from the investigator. The following measures are, therefore, proposed to support the interpretation of the final network solution: (i) the average influence (avi)[1] of a particular attribute x_i

$$\text{avi}(x_i) = \frac{1}{n}\sum_{p=1}^{n}\frac{\partial f}{\partial x_i}(\mathbf{x}_p) \tag{7.5}$$

[1]The partial derivatives are calculated numerically.

where f denotes the neural network solution; (ii) the average absolute influence (avai) of x_i

$$\text{avai}(x_i) = \frac{1}{n}\sum_{p=1}^{n} |\frac{\partial f}{\partial x_i}(\mathbf{x}_p)|; \tag{7.6}$$

(iii) an index indicating the degree of monotonicity of f in x_i

$$\text{mon}(x_i) = \frac{1}{n}|\sum_{p=1}^{n} I^+(\frac{\partial f}{\partial x_i}(\mathbf{x}_p)) - I^-(\frac{\partial f}{\partial x_i}(\mathbf{x}_p))|, \tag{7.7}$$

where $I^+(x) = 1$ if $x > 0$ and $I^+(x) = 0$ if $x \leq 0$, and $I^-(x) = 1$ if $x \leq 0$ and $I^-(x) = 0$ if $x > 0$. The avi(x_i)-measure indicates the average direction of influence of attribute x_i on MEDV, avai(x_i) indicates the average importance of attribute x_i in approximating MEDV –assuming that all inputs are scaled onto the same range, and mon(x_i) indicates the degree of monotonicity of the approximating function f in the attribute x_i. The latter measure can be regarded as one minus the proportion of sign-conflicting partial derivatives. A value of mon(x_i) close to one provides evidence for a monotonic underlying partial relationship, whereas a value close to zero provides evidence for a non-monotonic underlying partial relationship.

A plot of $\frac{\partial f}{\partial x_i}(\mathbf{x}_p)$ against x_{ip} for each attribute x_i may also help to interpret the final neural network solution. These plots indicate which attributes affect MEDV linearly and independent of other attributes; which attributes affect MEDV nonlinearly and independently of other variables; and which attributes affect MEDV in more a complex manner.

The avi(x_i), avai(x_i), and mon(x_i) measures per attribute are presented in Table 7.8. All input variables were scaled onto the same range, so the influence measures suggest which attributes affect housing price strongest. According to the avai-measures in Table 7.8 the main determinants of the house price are the house characteristics DIS, RM, LSTAT, NOX, and RAD. The signs of the corresponding avi-measures correspond with the direction of influence suggested in [HR78] and with the signs of the corresponding coefficients in the regression models (7.2) and (7.3). The last column in Table 7.8 indicates that the main determinants are in a (more or less) monotonic relationship to MEDV. A striking non-monotonicity (0.06) seems to exist between INDUS and MEDV.

Figure 7.2 provides additional information; it displays the partial derivative for each attribute in each observation of the sample as a function of the attribute's value. Relationships linear in the variables have constant derivatives everywhere. Quadratic or cubic dependencies have derivatives that behave as a linear or as a quadratic function respectively. Interactions among variables results in "scattered behaviour" of the partial derivatives to those variables. Figure 7.3 gives for comparison purposes similar plots for regression model (7.3).

Interactions between variables are clearly present. A striking difference between the neural network model and model (7.3) is that the partial derivatives of the neural network sometimes

Table 7.8: The avi(x_i), avai(x_i), and mon(x_i) measures.

attribute	avi(x_i)	avai(x_i)	mon(x_i)
CRIM	-0.07	0.08	0.62
ZN	-0.03	0.04	0.34
INDUS	0.03	0.11	0.06
CHAS	0.02	0.03	0.13
NOX	-0.23	0.23	1.00
RM	0.28	0.30	0.76
AGE	-0.09	0.10	0.87
DIS	-0.33	0.33	0.99
RAD	0.21	0.21	0.97
TAX	-0.18	0.11	0.98
PTRATIO	-0.16	0.16	0.98
B	0.05	0.05	0.79
LSTAT	-0.26	0.27	0.76

change signs, which indicates a deviation from the monotonicity assumption, often made in economics. An interesting example, suggested by the mon(x_i)-measure in Table 7.8, is the one between INDUS and MEDV. Figure 7.2 shows that above a certain quantity the level of nonretail business INDUS affects house price positively, whereas below this level it affects house prices negatively. A possible explanation could be that in areas with low business activity people are attracted by the pleasure of living (such as quietness, scenic environment, etc.), which diminishes when the level of industrial activity increases. Consequently, house prices are negatively affected by an increase in the level of industry. Living in an area with high business activity is attractive because commuting time is reduced to a minimum. When the level of business activity increases in these areas, the area becomes even more attractive to live in. House prices, consequently, are positively affected by an increase in the level of industry.

The foregoing has indicated that it is possible to analyse the "black box" neural network model, allbeit with some effort.

7.5. Conclusions

This case study illustrated that neural networks may well in the specification of regression models when there is no theory available that suggests a proper functional form. The neural network has found a specification of the hedonic house price model that fits the data better than

corresponding linear regression models, both in-sample and out-of-sample. The cross-sectional type of the data, the relatively large sample size, and the moderate number of relevant attributes makes the results practically useful.

Janssen [Jan92], who applied linear multiple regression techniques to specify the house price equation, concludes that his results might be improved by considering alternative model specification techniques, such as factor analysis, cluster analysis, and logit-regression. We state that neural networks should be considered as well, and may even be preferred above the other techniques. Hence, the lack of precise theories on hedonic models and the ample volume of data make data driven model specifications more promising than parametric modelling techniques. We have shown that in the case of the hedonic house price model for housing in Boston neural networks achieved a better model specification than the multiple regression technique. Since hedonic models can be constructed for many goods, we think that neural networks have good prospects for this particular area in economics.

The prediction performances of the three different models have been compared statistically by resampling based pairwise t-tests adjusted for the multiplicity effect. The tests justify the statement that the neural network model fits unseen data significantly better than the other two models.

Finally, we have made a first step towards the analysis of the final network solution. To this end, we proposed three measures: the first indicates the average direction of influence of a specific input on the network output, the second indicates the importance of a particular input in determining the output, and the third measure indicates the degree of monotonicity in the partial relationship. Visual inspection of the plots of the partial derivatives of the final network for each network input in each sample observation provides another useful aid in the analysis of the final neural network solution.

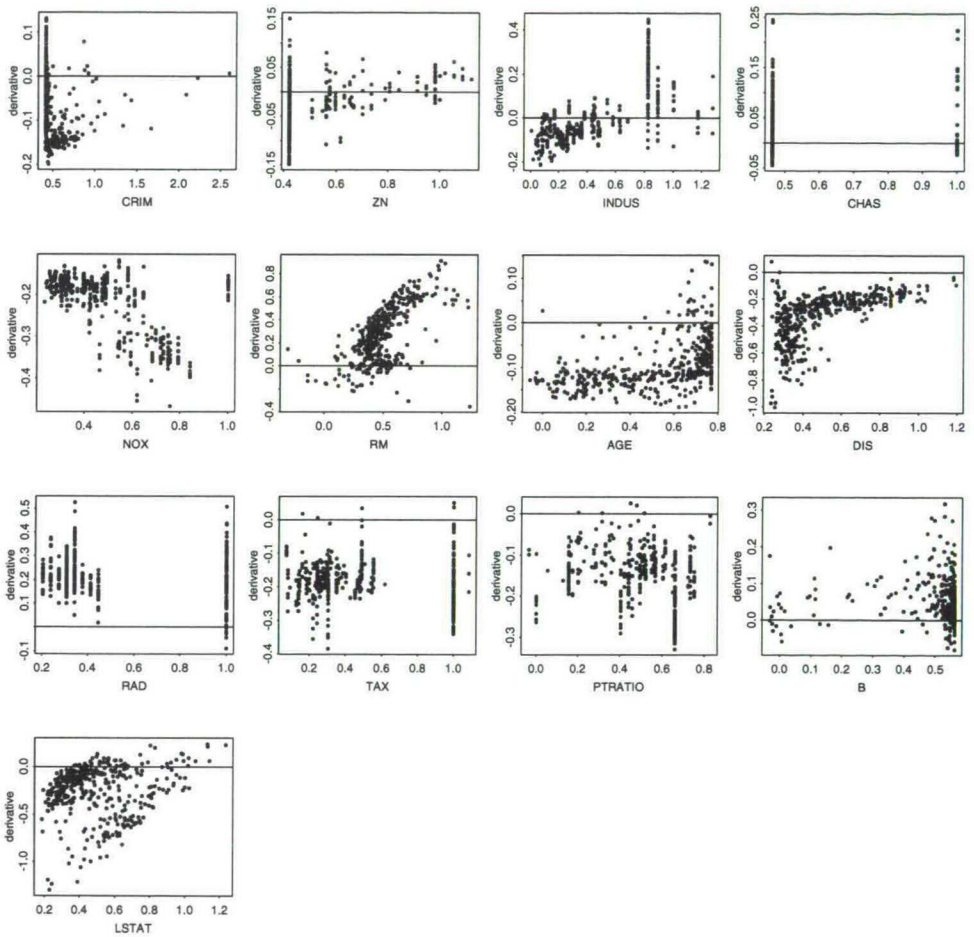

Figure 7.2: Partial derivatives for each attribute in the NN model

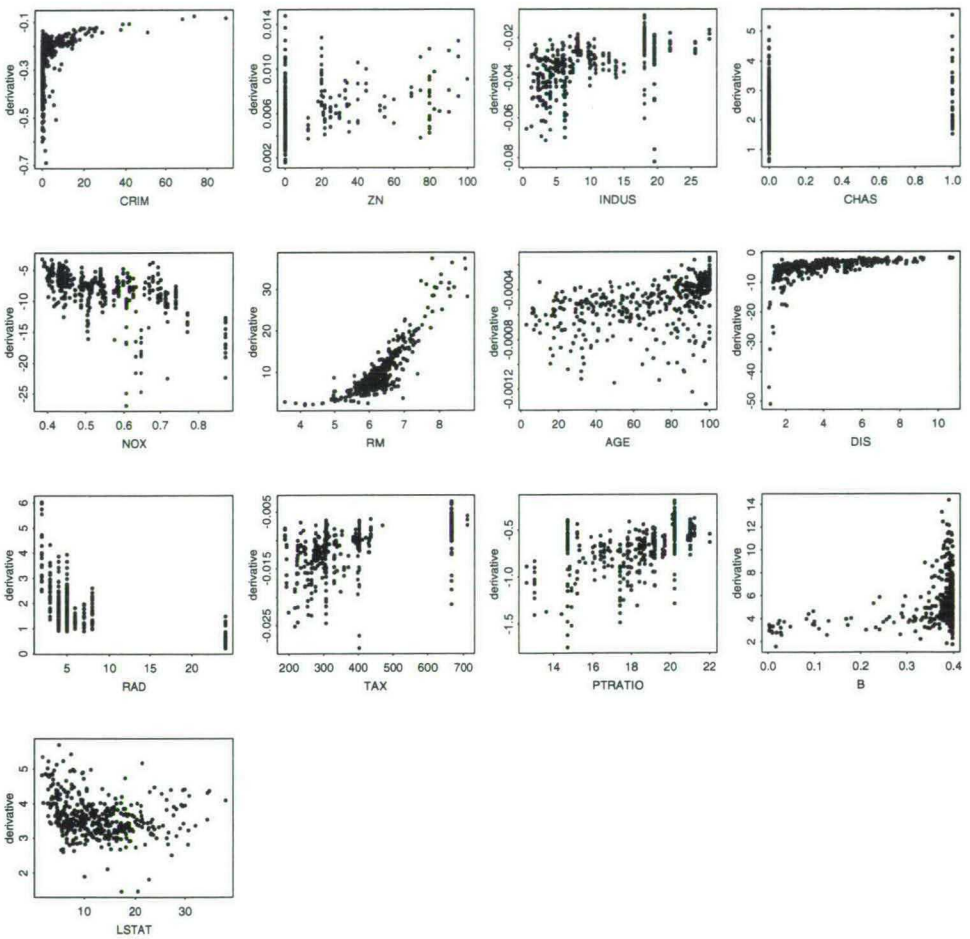

Figure 7.3: Partial derivatives for each attribute in the linear model (7.3)

Chapter 8

Predicting the Dutch Mortgage Loan Market

8.1. Introduction

Mortgage lending[1] is an important activity for banks, which generates a large part of their profits. The total amount of outstanding mortgage loans constitutes a main component of the bank's balance sheet. There is a need for making accurate predictions of the future path of the stock of mortgages, since it must be funded with borrowed money. Hence, an unexpected increase in the stock of mortgages asks for an immediate attraction of large sums of funding money, usually at prices higher than the average market value at that time.

The traditional approach to such economic modelling problems is multiple regression. The field of Artificial Intelligence, however, has provided a battery of data modelling methodologies [TG91], including neural networks, expert systems, case based reasoning, decision trees, and qualitative reasoning, which also apply to economic modelling in general. These techniques use approximations to the underlying data generating mechanism that are more flexible than parametric regression models.

The interest of the study is to obtain the best possible predictions for the production of new mortgage loans. The parametric econometric approach, which is conventionally used for such problems, is adopted as a "bench mark" for the AI-techniques.

[1]This chapter is largely based on Verkooijen and Daniels [VD95], which is conducted within an EC-funded network of the SPES programme (contract number 0065). The SPES (Stimulation Plan for Economic Science) project, entitled "Artificial Intelligence approaches to modelling in Economics", is a joint research project with participants from Heriot-Watt university (United Kingdom), Tilburg University (The Netherlands), Milan University (Italy), ABN/AMRO bank (The Netherlands), and Digital Equipment Europe (France). The ABN/AMRO bank proposed "modelling of the Dutch mortgage loan market" as joint problem for the participating partners in the SPES project.

The outline of the chapter is as follows. In section 2 we will summarise Asset and Liability Management (ALM) and its relation to the mortgage market. Section 3 characterises the Dutch mortgage market. In section 4 several UK studies on mortgage demand are reviewed. Section 5 departs from simple economic theory, in order to arrive at a demand function for mortgage borrowing that will be the basis for the empirical model. Section 6 describes the design and the results of the empirical study that employs neural networks to build an error-correction model for the production of new mortgage loans in the Netherlands. Section 7 concludes the chapter.

8.2. Asset and Liability Management

Banks improve the efficiency of financial markets by their acting as financial intermediaries. These activities create profitable opportunities, but may also introduce risks for the bank. Interest rate risk is an important determinant. Interest rate risks originates from borrowing and issuing money at fixed interest rates at different (unmatched) maturity terms. Loans with different maturity terms have different interest rates, graphically represented in the yield curve. In general, the yield curve increases monotonically, i.e., the interest rate for short-term loans is generally below the interest rate for long-term loans, since the latter incorporates a higher risk premium; nevertheless, for short periods of time the short-term rate may lie above the long-term interest rate (inverse yield curve).

Banks generally make profits by issuing money at an interest rate higher than the rate at which it is funded. This is called interest profit; interest rate profit forms the main component of the bank's income (in 1994 it constituted 65% of the total assets of the ABN/AMRO bank). The maturity terms of issued and funding loans typically do not match; money issued for a long-term period (say 5 years) is funded with money lended for a short period (say 3 months). This mismatch causes the interest rate risk, since short-term interest rates may rise in the future. When the short-term interest rate rises beyond the rate at which long-term money was issued, it may eventually cause an interest rate loss (at least partly).

In general, the value of issued loans is higher than the value of funding money. In order to control the level of interest rate risk, the Asset and Liability COmmittee (ALCO) instructs the Treasury department to make up the gap between issued and funded money. In doing this, the Treasury should keep the costs as low as possible by carefully selecting profitable types of funding. ALCO provides the Treasury department with instructions concerning the term structure of the money to be issued and borrowed. The ALM department supports ALCO in its policy by making simulations of the future balance sheet. The issuing of long-term loans is not controlled by the bank, but is mainly determined by the market. The major task of the Treasury department, therefore, is to select the cheapest funding money.

In performing their task, the Treasury department has to rely heavily on predictions, since

the actual values of most transactions are available only after several months. Therefore, it is crucial to have accurate predictions of all, or at least the main, components of the balance sheet. Otherwise, the Treasury department may have to borrow large sums of money at prices higher than necessary, in order to immediately correct for an unanticipated gap between assets and liabilities.

A major component of the bank's balance sheet is the portfolio of mortgage loans. In the remainder we will focus on this component exclusively. The bank requires a model that predicts the value of new mortgage loans one month ahead, up to 18 months ahead.

8.3. The Dutch Mortgage Market

A mortgage loan is a loan that has a property acting as security for the fulfilment of the borrowers' obligations. In case the obligations are not fulfilled, the lender has the right to let the collateral (security) be sold in public to fulfil the claim.

The mortgage loan market is quite diverse, on both the supply and the demand side. Not just one type of mortgage loan is supplied, but a whole set of different mortgage loans which differ in the maturity period, the repayment conditions, the amount of money that can be borrowed, the interest rate, and the period over which the interest rate is fixed.

There are a number of institutions that provide mortgages, among them are general banks (ABN/AMRO, ING, RABO), mortgage companies, savings banks, insurance companies, building societies, and pension funds.

In the Netherlands the supply of mortgages can be regarded as perfectly elastic in practice, that is, banks will provide a requested mortgage loan after the applicant has passed a general check for creditworthiness. Pau [PT90] developed a knowledge-based system that automates the credit granting process. So, the provided number and value of mortgage loans can safely be regarded as completely determined by demand. Consequently, in contrast to other countries (for instance, the United Kingdom), rationing mechanisms can be neglected in the Netherlands. In 1991 the limit on the maximum amount of borrowing was increased by making the maximum amount of borrowing dependent on the incomes of both partners instead of the income of the highest earning partner.

The demanders for mortgage loans can be distinguished into four groups:

- *starter* buys a house for the first time,
- *mover* moves from his current privately owned dwelling to another privately owned one,
- *raiser* has not moved, and has not changed his mortgage institution, but only raised his mortgage level,

- *changer* has not moved, but changed, for instance, from one mortgage institution to another, or from a mortgage loan at a high rate to a mortgage loan at the current (lower) rate.

Demand per group reacts differently to changing (economic) factors.

The sequel gives a picture of the Dutch mortgage market in 1992 (from [Kie93]). A total of 243,260 new mortgages were registered in the land register. In 80% of these cases the collateral was a house (finished or under construction). The largest part (90%) of the total mortgage market was determined by mortgages taken out on houses; when calculated in values, this percentage drops to 67%. In 1992, 76% of the total of new mortgages corresponded to a "mover" (34%) or a "starter" (66%). Of the 24% that did not move or start, 12% bought their rented house, and 79% switched between mortgage institutions.

The general banks accounted for 45% of the new mortgages loans, the mortgage companies for 15.4%, the assurance companies for 13.3%, the building companies for 7.8%, the savings banks for 7.5%, and the remaining suppliers for 5.6%. Contrary to, for instance, the UK, in the Netherlands building societies play only a minor role on the supply side of mortgage loans. The RABO-bank owns the largest share of the mortgage market (15.6%, measured in Dfl.), followed by the ABN/AMRO (11.6%).

Of the 80% new mortgages on housing 27% was used for newly built houses. Private individuals own 45% of the total housing stock, which is expected to increase to 55% in the next 10 years (see [HJKS92]).

The next section will review some UK and US studies on the demand for mortgage loans. In the empirical part the assumed determinants of the mortgage loan demand are examined for the Dutch case.

8.4. A Survey of Previous Studies on Mortgage Markets

8.4.1. United Kingdom

The largest part of the literature on mortgage markets concerns the United Kingdom. In many UK studies ([Had76, May79, Hol92, AH84]), the demand for mortgages by private households often appears as part of a larger econometric model of building societies. "United Kingdom Building Societies are non-profit-making financial intermediaries ... They dominate the UK mortgage market with their assets representing about an 80 per cent share of housing finance ..." [AH84]. This was particularly the case during the 1970s [HU89].

Building societies could, by a cartel arrangement, offer their clients interest rates that followed a stable time path. A consequence was that a rise in market interest rates could not immediately be followed by the building societies' rates. The building societies saw their

inflows decline accordingly. After some time, shortage of funds forced societies to reduce the rate of growth of the mortgage stock. Since the societies would not raise interest rates by an amount high enough to restrict mortgage demand sufficiently, they relied on a variety of non-market mechanisms, including borrowers queueing for mortgages and changes in lending arrangements, such as by lowering the ratio of loan to property value or loan to income [HU89]. In the literature, these mechanisms are called 'rationing' [Wil85].

In mid-1980 the system of direct controls on the banks' lending ended, which gave the banks greater freedom to develop new areas of business activity. In particular, banks focussed on the personal sector. To day, the UK mortgage market has become more competitive, which implies a greater sensitivity to changes in market interest rates [HU89].

The following paragraphs review the literature on modelling demand for mortgages. As said before, most demand equations appear as part of a larger system of equations.

Hadjimatheou [Had76] assumed the demand for building society mortgage advances to be determined by personal income, real cost of borrowing (measured by the mortgage rate of interest adjusted for tax relief), rate of change of new house prices, relative price of new housing, number of marriages, lagged value of building societies advances, and seasonal dummies. The number of marriages is used in place of new household formation, for which quarterly data were not available. The complete equilibrium model was based on seventy-five quarterly observations corresponding to the period running from the second quarter 1955 to the fourth quarter 1973. Five additional observations were used for a post parameter stability test.

Mayes [May79] states that the maximum supply of new mortgage advances without constraints would clearly be the demand, so that a specification of the demand for mortgage advances would be a suitable first step. Mayes assumes this demand to be a function of real personal disposable income, the price of the mortgage (i.e., the mortgage rate), the price of housing relative to consumer prices in general, and some seasonal factors. To this set of determinants, he added some factors that constrain the demand (liquidity ratio and reserve ratio).

Martin and Smyth [MS91] assume that the borrower's demand for real mortgage loans is a function of the rate of interest, real permanent income, the rate of inflation in homeowner's cost (in the previous period), the rate of inflation in renter's cost (in the previous period), and the real purchase price of housing. The model was estimated in double log form on monthly observations covering the period from June 1968 through March 1989.

In his econometric model for the building societies, Pratt [Pra80] represents the demand for net building society advances by the inflation in new house prices, the households' disposable income, the inflation in consumer prices, the change in the real (after tax) mortgage rate, and the change in the percentage change over the last six months in the relative price of housing. The last factor was included to capture the 'speculative' demand for housing. The study used quarterly data from 1966 through 1978.

Wilcox [Wil85] assumes the demand for the stock of building society mortgages to be determined by real personal disposable income, the consumer expenditure deflator, the house price index, the average mortgage rate net of the basic rate of tax, the loan-to-value ratio, and the value of the owner-occupied housing stock. The parameters in all equations were estimated from quarterly data for the period 1968 through 1984.

Anderson and Hendry [AH84] formulate a disequilibrium theory for Building Societies' behaviour. Their econometric model assumes that the personal sector demand for mortgages may be determined by the real disposable income, the price level, the rate of inflation, the after tax mortgage rate of interest, the price index of a 'standardised' house, and the house price inflation. They expect that each factor –except the mortgage rate of interest and the house price– positively affects the demand for mortgages, that the mortgage rate of interest affects the demand negatively, and that it is unclear in which direction the house price affects the demand. The model was estimated on quarterly observations from the period 1958 through 1979.

Hall and Urwin [HU89] formulate and estimate an explicit disequilibrium model of the supply and demand for mortgage lending over the period between 1970 and 1985. They assume the demand for real mortgage lending to be determined by the rate of interest on mortgages, the relative price of houses, real disposable income, the general price level of goods, and the number of owner occupied houses. The study used quarterly data.

Holmes [Hol92] focussed on the demand for building society mortgage finance in northern Ireland and Scotland. Holmes, as well as many of the studies discussed above, assumes that the demand for mortgage is subjected to a "partial adjustment" mechanism, that is, mortgage demand in the previous period has only be partially met. The rationale is that households face major adjustment costs when adapting to a new desired level of mortgage borrowing. Besides the mortgage demand in the previous period, Holmes assumes the following determinants of the demand for mortgage loans: nominal gross domestic product, the average price of housing, and the real mortgage interest rate adjusted for tax relief. The study is based on annual observations for the period 1970 through 1989.

8.4.2. The Netherlands

There are surprisingly few studies performed on the Dutch home mortgage market. The Dutch National Bank (DNB) has a quarterly model for the Dutch economy, called MORKMONII [FKB90], which includes an equation for the long-term withdrawal of assets by households ("lang opgenomen middelen door gezinnen") LOG. The main part of LOG concerns mortgage loans. Therefore, the equation incorporates the value of the stock of owner occupied houses SH as a "scaling" variable. Additionally, the long-run (nominal) interest rate rl and the ratio of LOG and SH are included as explanatory variables. Besides the demand factors, a credit-restricting

dummy Dum is included. The estimated equation, reported in [FKB90], is

$$\begin{aligned}\Delta LOG_t &= 0.0135\, SH_t - 0.0018\,(0.5\,(rl_{t-1} + rl_t)\, LOG_{t-1}) \\ &\quad -4296.4\,(LOG/SH)_{t-1} - 0.0045\, Dum\, LOG_{t-1}. \end{aligned} \tag{8.1}$$

The equation is estimated using data from the first quarter of 1971 through the fourth quarter of 1987.

The macro-econometric model for the Netherlands [Bur92], which has been developed by the Central Planning Bureau (CPB), also incorporates a part concerning the stock of privately owned houses and the stock of mortgages. However, correctly isolating this part from the total model is not so straightforward, especially due to the poor documentation on this particular part.

Since the housing market is very closely linked to the mortgage market (as we saw in section 8.3), it may be helpful to build a model for this market as well. Two recent studies, which elaborate on developments in the Dutch housing market, are Janssen [Jan92] and Hut *et al.* [HJKS92]. Janssen [Jan92] focusses on the theory on house price determination, and provides empirical results for four cities (Eindhoven, Enschede, Lelystad, Rosmalen). Hut *et al.* discuss long-run structural developments in the Dutch housing market. Both studies provide valuable information in future.

All studies mentioned so far presume a strong connection between the housing market and the mortgage market. Jones [Jon93] modelled home mortgage demand in a manner that separates that demand into the amount derived from housing demand and the demand from financing nonhousing assets. The Dutch National Bank recognises this distinction in its annual report of 1994; to explain the observed (further) increase of the amount of outstanding home mortgage debt by households despite an increasing mortgage interest rate, they argue that a larger part of the credit taken out on dwellings is used for consumer expenditures. Mortgage loans are cheaper than consumer credit, and the strongly increased value of the collateral (dwellings) makes that requests for additional mortgage loans are more easily approved.

In summary, the following factors are assumed to determine the long-run demand for mortgage loans:

1. household's disposable income (real or nominal),
2. mortgage rate (nominal or adjusted for tax relief),
3. price (value or inflation) of new or existing housing,
4. (general) price inflation,
5. costs of renting,
6. costs of house ownership,
7. number of weddings (new household formation),
8. value of the housing stock.

8.5. A Demand Equation for Mortgages

In the previous section we summarised the determinants of the demand for mortgage borrowing which were used in several UK studies that –explicitly or implicitly– modelled the demand for mortgage loans. In this section we derive a reduced form equation for the demand for mortgage, based on simple economic principles.

The demand for mortgage loans is derived –after [HU89]– from a simple utility maximisation principle. Suppose a representative household has the utility function $u(H, G)$, where H represents housing services and G an aggregate of other goods. This household will then try to maximise this utility function subject to a constraint imposed by the disposable income of the following form:

$$g_1(r_m, P^H)\, H + P\, G = Y_d, \tag{8.2}$$

where $g_1(r_m, P^H)$ is a (simple) cost function of servicing a mortgage which will provide housing services H, P is the general price level of goods, and Y_d is the household's nominal disposable income. The cost function g_1 is assumed to depend on the mortgage interest rate r_m and the price of houses P^H. This will yield a general constrained demand function for housing services of the form:

$$H = g_2(r_m, P^H, Y_d, P). \tag{8.3}$$

The foregoing analysis neglected the decision of home ownership against renting. Some factors governing this decision are, for instance, the relative cost of renting opposed to the cost of home ownership. These factors could be included in (8.3). However, it may be argued that the main changes in owner occupation in the Netherlands over the last 20 years have been due to institutional factors (sale of rented houses and rate of release of building land) more than to purely economic factors. Therefore, it may be preferable to scale equation (8.3) by the number of owner occupied houses (say) NOH, to derive a desired *aggregate* demand for mortgage M^d [HU89]:

$$M^d = g_3(r_m, P^H, Y_d, P) \cdot NOH. \tag{8.4}$$

Households face enormous adjustment costs when changing the level of their mortgage borrowing (in many cases it involves moving to another house or building a considerable extension to a house). Therefore, it seems reasonable to assume a partial adjustment process to the desired level of mortgage borrowing, that is,

$$M_t = M_{t-1} + \gamma(M_t^d - M_{t-1}) + \epsilon_t \quad (0 < \gamma \leq 1), \tag{8.5}$$

where M denotes actual mortgage borrowing and M^d desired mortgage borrowing. Thus, (8.5) asserts that in the current period the households only move part of the way towards the optimal level of mortgage borrowing, the speed of adjustment being determined by the parameter γ. Note that an error-correction model might be used instead of the partial adjustment model.

The mortgage advances $A_t := M_t - M_{t-1}$ are then a function of the following form:

$$A_t = \gamma \left(g_3(r_m, P^H, Y_d, P)_t \cdot NOH_t - M_{t-1} \right) + \epsilon_t. \tag{8.6}$$

The mortgage advances can also be defined as the inflow of new mortgages (the mortgage production) minus the outflow of mortgages:

$$A_t := M_t^{in} - M_t^{out}. \tag{8.7}$$

Since the ABN/AMRO bank is merely interested in predicting M^{in} and we have monthly observations on M^{in} only at our disposal, we are forced to model the inflow in the mortgage stock instead of the mortgage stock advances, as is usually done in the literature ([HU89, AH84, Pra80]).

We assume that a constant proportion of the outstanding stock of mortgages is ended, which is a very simple model for the outflow of mortgages. This assumption combined with (8.6) and (8.7) yields the following reduced form equation for the production of new mortgage loans:

$$M_t^{in} = g_4(r_{m,t}, P_t^H, P_t, Y_{d,t}, NOH_t, M_{t-1}) + \nu_t. \tag{8.8}$$

If the outflow of mortgages is a decreasing function of r_m (which is the case when people on mortgages with a high fixed rate switch to loans at the lower current mortgage rate), the same reduced form equation (8.8) is implied.

The derived reduced form (8.8) clearly depends on the assumptions we have made. The procedure merely illustrates how a demand function for mortgage borrowing can be derived theoretically. If other assumptions were made (for instance, instantaneous adjustment to the desired level of mortgage borrowing), a different reduced form would result. The reduced form equation (8.8) will be used in section 8.6 to specify an estimable long-run model for the production of new mortgage loans.

8.6. Empirical Study

As indicated earlier in section 8.1, our objective is to develop a model for predicting the *production of new mortgages* (measured in millions of Dutch guilders) for 1 month ahead, up to 18 months ahead. In such situations it is common practice to build different models for different horizons: short-run models, which concentrate on rapidly changing variables (measured on a monthly basis), and long-run models, which are based on slowly moving variables (measured on an annual basis). The two models may well have quite different specifications with non-overlapping sets of explanatory variables.

We synthesise both long-run and short-run models to obtain a prediction model for the Dutch mortgage market that suits our purposes. Engle *et al.* in [EGH91] propose a strategy to

combine short-run and long-run aspects of the data generating process (measured at different time frequencies) into a single error-correction model. Our approach is largely based on this strategy.

The deviation of the observed mortgage production from the long-run model's predicted value in the previous period is incorporated as explanatory variable in an error-correction model of the form:

$$\Delta M_t^{in} = \alpha_0 + \psi(V_t) - \gamma\, ecm_{t-1} + \nu_t \quad (\gamma > 0) \tag{8.9}$$

where V_t denotes the stationary short-run variables, ψ a yet unspecified functional form, ecm_{t-1} the deviation of M_{t-1}^{in} from its predicted long-run value at time $t-1$, and ν_t a noise term. Since the long-run model uses annual data, its predictions are interpolated to obtain monthly forecasts.

Traditionally, the function ψ in (8.9) is assumed to be linear: $\psi(V_t) = \alpha^T V_t$. An innovating aspect in our study is that the linear setting is enlarged by allowing for possible nonlinear specifications of ψ. Neural networks are used to explore for possible nonlinear specifications (see also Chapter 5).

We build a long-run model for the production of new mortgage loans, using annual data. Because of the limited length of the annual series (30 observations), a simple linear model, loosely based on (8.8), is specified and estimated. Neural networks are not considered for the specification of the long-run model, since the data set is too limited to justify a nonlinear relationship.

Introducing nonlinearities in the error-correction model, however, may improve its specification. Economic theory usually has not much to say about the particular form of the short-term nonlinear dynamics. Therefore, neural networks are used to search for possibly nonlinear specifications of the error-correction model.

8.6.1. The Long-run Model

Because we have at maximum 30 yearly observations at our disposal, we estimate a simple long-run model. The factors outlined in the reduced form equation (8.8), complemented with the factors outlined in section 8.4, are considered as candidates for inclusion in the long-run model.

The characterisation of the limited set of annual data as $I(0)$ or $I(1)$ (see section 5.3) by unit root tests may easily lead to wrong conclusions, since unit root tests are meant for large samples. Therefore, we will refrain from performing such tests. A visual inspection of the autocorrelation plot of the variables listed in Table 8.1 suggests that they are all $I(1)$. Table 8.1 presents the definitions of part of the available data series; more information is found in Appendix B.

A disadvantage of modelling the *production* of new mortgage loans instead of the net *advances* in the stock of mortgage loans, is that borrowers who change their mortgage contracts,

Table 8.1: Symbols and names of long-run variables.

variable	name
M^{in}	production (value) of new mortgages
M	stock of mortgage loans
Y_d	households' real disposable income
r_m	mortgage loan rate (5 years)
P^H	price of housing
NOH	number of owner occupied houses
wed	number of weddings
inf_p	consumer price inflation
inf_r	inflation in rent prices

but not raise its level (switch to another lender or switch to a cheaper loan) add to the production, but not to the advance in the stock of mortgage loans. We conjecture that in 1993 and in 1994 in particular, these changers are responsible for a considerable proportion of the mortgage production in those years; the reason is twofold. First, the mortgage rate dropped from a yearly average of approximately 9.5% in the period 1990-92 to a level of approximately 7.5% in 1993 and 1994, so the decline is large enough to compensate for the fine that changers have to pay for early repayment of the loan. Second, in the recent past, mortgage bureaus have entered the market, pointing borrowers at the potential cost reductions they will meet when they change their mortgage loan to one with a lower rate (including the fine for early repayment of the loan). Modelling this part of the production of mortgage loans is difficult, since the borrower's awareness of these profitable possibilities has evolved over time, and no representative historical cases are available yet.

Based on the reduced form equation (8.8) with the economic ideas behind it, and on model (8.1) developed by the Dutch National Bank, we arrive at the following empirical long-run model:

$$M_t^{in} = \beta_0 + \beta_1 NOH_t P_t^H + \beta_2 r_{m,t} M_{t-1} + \gamma M_{t-1}. \tag{8.10}$$

This model is estimated from data for the period 1965-94. The results (coefficients, t-values, and R^2) are shown in the Table 8.2. The addition of one or more of the variables from section 8.4 does not improve the fit considerably. So, we decided to keep the specification of the long-run model simple.

Figure 8.1 depicts the observed production of new mortgages and the predictions based on (8.10), for the period 1966-94.

Table 8.2: Estimation results for model (8.10).
1965-1994

variable	coeff	std. error	$t-value$
intercept	-3305	1129	-2.93
$NOH_t \cdot P_t^H$	0.1886	0.0098	19.27
$r_{m,t} \cdot M_{t-1}$	-0.6477	0.1650	-3.92
M_{t-1}	-0.0405	0.0154	-2.63
R^2	0.977		
DW	1.23		

8.6.2. The Error-correction Model

We have monthly observations on most (potentially) relevant variables from January 1985 through December 1994 at our disposal. So we model the production of new mortgages on a monthly basis; see Appendix B for the details. It is evident that other (economic) factors than in the yearly model will explain the monthly changes in the production of new mortgage loans.

Figure 8.2 depicts the monthly production of new mortgages from January 1985 through December 1994. Two striking characteristics are the seasonal pattern (January low, December high, and an intermediate peak in July and August) and the strong increasing trend. The trend

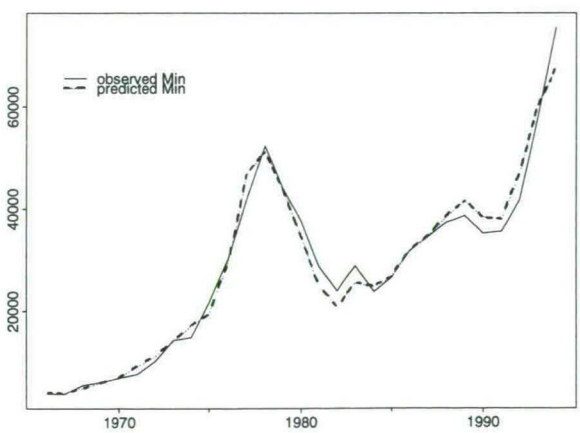

Figure 8.1: The fit of the long-run model

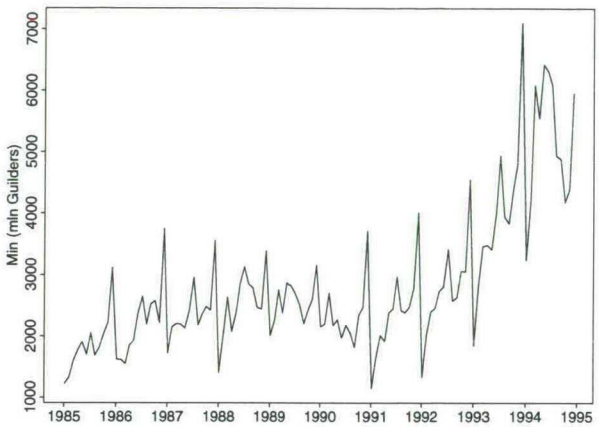

Figure 8.2: Monthly production of new mortgage loans

component is captured by the long-run model (8.10). Figure 8.3 shows that the seasonal pattern is likely due to the consumers' house buying behaviour. The first window of this figure displays

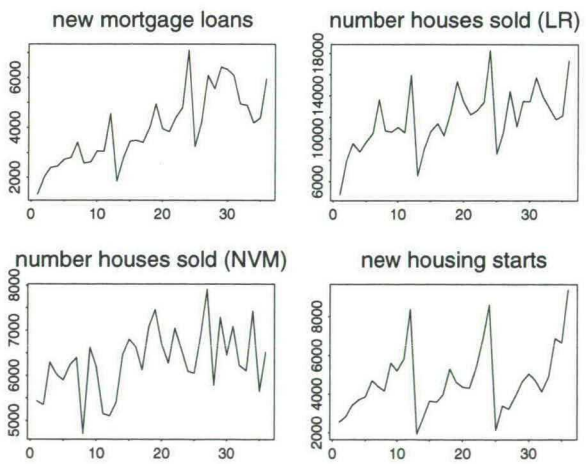

Figure 8.3: Seasonal pattern of the production of mortgage loans

the production of new mortgages during 36 months (Jan. 1992 to Dec. 1994); the second window displays the total number of houses sold in the same months (measured by the land's register);

the third window displays the total number of houses sold by NVM agents (Dutch Society of real estate agents); and the fourth window displays the new housing starts in the same period. Three series more or less have the same seasonal pattern, except for the NVM data.

However, the data on the number of houses sold as measured in the land's register are more reliable than the data from the NVM, in which the seasonal pattern is less apparent. The data from the land's register are available only from January 1992 onwards, whereas the NVM data range from January 1985 until December 1994. Therefore, we use the NVM data to specify and estimate the error-correction model. Seasonal dummies are included to correct for the lack of seasonality in these data.

Besides the trend and the seasonal component, changes in the monthly mortgage rate and in the nominal value of houses may influence households' decisions to take out mortgages in a particular month as well. The error-correction model comprises the following variables in addition to three[2] lags of the dependent variable ΔM_t^{in}:

- $\Delta(H^s \cdot P^H)_t$ the change in the nominal value of existing houses sold (up to three lags),
- $M(-1)_t \, \Delta r_{m,t}$ the change in the mortgage rate (up to three lags) scaled by the amount of outstanding mortgages at the end of the previous year,
- ecm_{t-1} the error-correction term, that is, the lagged deviation of the observed ΔM_t^{in} from its value predicted by the long-term model,
- seasonal dummies.

In contrast to standard error-correction models, which use data of the same frequency, our error-correction model uses both annual and monthly data. The annual predictions (divided by twelve) are transformed into monthly predictions by a cubic spline[3] (following [EGH91]). The error-correction term links the monthly movements in the production of new mortgage loans to the long-run production.

8.6.3. Model Selection

We employ neural networks to arrive at an empirical specification of the error-correction model (8.9), in particular to specify its ψ-function. In Chapter 3 the mathematical representation of a single layer feed-forward neural network with one linear output unit and skip-layer connections was introduced. We therefore parametrise equation (8.9) by a neural network as follows:

$$\Delta M_t^{in} = \alpha_0 + \{\alpha^T V_t + \sum_{i=1}^{N_h} \beta_i \phi(\mathbf{w}_i^T V_t)\} - \gamma \, ecm_{t-1}, \tag{8.11}$$

[2]The number of lags is chosen on purely pragmatic reasons.

[3]We placed the knots at each June.

where ϕ represents the neural network's sigmoid squashing function, N_h the number of hidden units; the other symbols were introduced earlier. The parameters are simultaneously determined by minimising the squared error function with penalty term λ (see Chapter 3, equation 3.9). This λ is used as smoothing parameter; smoother approximations are obtained for larger values of λ.

All empirical specifications of the error-correction model which we consider are subsumed in (8.11). The standard linear form is simply obtained by fixing N_h at zero (a neural network without hidden units). Increasing the number of hidden units enlarges the class of functions that can be approximated by the neural network; a nonlinear relationship may be revealed, if present.

V_t in (8.11) represents the set of short-run variables which are presumed to be relevant when predicting ΔM_t^{in}. Three encompassing sets of variables V_t are examined. The error-correction form (8.11) that corresponds to each set of variables is referred to by m_i ($i = 1, 2, 3$). The first model, m_1, simply includes all variables (and their lags) selected in section 8.6.2, eleven seasonal dummies, and an intercept term. The second model, m_2, includes only those variables from m_1 which have significant coefficients at an error level of 5%, when the linear specification of m_1 is estimated by OLS. The third model, m_3, includes only ecm_{t-1} and the seasonal dummies which were significant in model m_1.

Table 8.3: Estimation results for the short-run model m_2.

variable	value	st. error	t-value
ΔM_{t-1}^{in}	-0.114	0.053	-2.17
$M(-1)_t \, \Delta r_{m,t}$	0.120	0.031	3.92
$M(-1)_t \, \Delta r_{m,t-1}$	-0.083	0.032	-2.63
$M(-1)_t \, \Delta r_{m,t-3}$	-0.083	0.028	-2.99
$\Delta(H^s \cdot P^H)_t$	0.072	0.032	2.24
ecm_{t-1}	-0.161	0.052	-3.07
jan	-0.724	0.058	-12.58
mar	0.111	0.050	2.21
jul	0.142	0.042	3.39
aug	-0.142	0.044	-3.19
dec	0.514	0.044	11.60
R^2	0.865		
DW	2.418		

Table 8.3 displays the variables included in model m_2, the coefficient values, the corre-

sponding standard errors, and t-values when m_2 is specified linearly. The seasonal dummies are present with the strongest effects for January (low) and December (high), as could be expected (see figures 8.2 and 8.3). The error correction term ecm_{t-1} is significant and has the correct negative sign, which implies that deviations from the long-term model are corrected for in the short-run. The changes in the number of houses sold (by the NVM) have also correct signs; an increase in the number of existing houses sold leads to an increase in the production of new mortgages. Several lags of the monthly changes in the mortgage rate are significant, and so are the changes in the monthly mortgage production one period lagged. The total effect of a change in the mortgage rate on the mortgage production is negative, which corresponds to what is expected from economic theory.

The following neural network analysis is performed to examine whether nonlinear specifications of the models m_1 and m_2 are justified by the data. Each model m_i is estimated under all possible combinations[4] of

$$N_h \in \{0, 2, 4\} \text{ and } \lambda \in \{5, 0.5, 0.1, 0.05, 0.01, 0\}.$$

The final model is selected on the basis of the cross-validation goodness-of-fit value (R^2_{cv}).

The results of the 10-fold cross-validation procedure[5], which was described in section 4.8, are summarised in Table 8.4. Its cells in Table 8.4 show for each model the within-sample coefficient

Table 8.4: Out-of-sample model comparisons.

model	N_h	weight decay value λ				
		5	0.5	0.05	0.01	0
m_1	0	0.816/0.732	0.885/0.794	0.887/0.783	0.888/0.780	0.888/0.780
	2	0.816/0.732	0.885/0.794	0.950/0.814	0.973/0.667	0.983/0.347
	4	0.816/0.732	0.885/0.794	0.951/0.780	0.989/0.549	0.999/0.173
m_2	0	0.793/0.719	0.863/0.800	0.865/0.803	0.865/0.803	0.865/0.803
	2	0.793/0.719	0.863/0.800	0.909/0.828	0.940/0.796	0.944/0.295
	4	0.793/0.719	0.862/0.800	0.909/0.821	0.955/0.768	0.971/0.320
m_3	0	0.758/0.721	0.820/0.778	0.820/0.778	0.820/0.778	0.820/0.778

note: cells display R^2_{in}/R^2_{cv}

of determination R^2_{in} and the out-of-sample coefficient of determination R^2_{cv}, calculated from the predictions made within the cross-validation procedure.

[4]The specific set of combinations has been chosen after some preliminary experiments.

[5]Observations corresponding to each year are consecutively left out.

The rows with $N_h = 0$ (a neural network without hidden units) present results for models m_i specified linearly and estimated by restricted OLS. When the weight decay value is set at zero, the results correspond to linearly specified models m_i with coefficients estimated by unrestricted OLS. When the weight decay value is larger than zero, the coefficient estimates of the linear model are no longer unbiased (ridge regression).

Nonlinear specifications of m_i may be found when N_h is larger than zero, of course running the risk of overfitting the data at hand. The low R^2_{cv}-values in combination with (very) high R^2_{in} at low values of the weight decay parameter when two or four hidden units are included, is a clear indication of overfitting. Within-sample the models fit the data (almost) perfectly ($R^2 > 0.99$), whereas out-of-sample the models fit badly ($R^2 < 0.30$). In these situations, higher R^2_{cv} (better out-of-sample fit) at higher values of λ illustrate how effectively weight decay reduces overfitting.

Table 8.4 further shows that model m_2 provides the best cross-validation results, and that a slight advantage is achieved when a neural network with two or four hidden units and a weight decay value of 0.05 is trained on the data.

8.6.4. Predictions

A model is selected to make real out-of-sample predictions for 1-18 months ahead (from July 1993 to December 1994). First we select model m_2 with $N_h = 0$ and $\lambda = 0$, and with parameters estimated on data until June 1993. The exogeneous variables are assumed to be perfectly predictable for the period July 1993 to December 1994, so we inserted observed values for them. The predictions are generated iteratively by equation (8.11) for model m_2; in each iteration step the error-correction term ecm_{t-1} is calculated by the previously predicted value $\hat{M}^{in}_{t-1}$ minus its prediction from the long-run model.

Figure 8.4 shows the out-of-sample predictions of the monthly production of mortgage loans, the observed production of mortgage loans, and the long-run predictions. It is apparent that the quality of the predictions, especially for mid 1994, is not so good. This is what we expected from the unequalled high proportion of mortgage changers in that period. Nevertheless, the strategy of combining long-run and short-run aspects of the data generating process into a single error-correction proves useful when making predictions for different horizons. Hence, these predictions are better than predictions made from the long-run model alone or from a short-run model which neglects the long-run trend.

Next we also generate predictions using model m_2 with $N_h = 2$ and $\lambda = 0.05$, which showed best results in Table 8.4. The resulting predictions were almost identical to the ones obtained by the linear specification of m_2 (with $N_h = 0$, $\lambda = 0$). Consequently, a plot of the predictions would not be discernible in Figure 8.4, so we left it out.

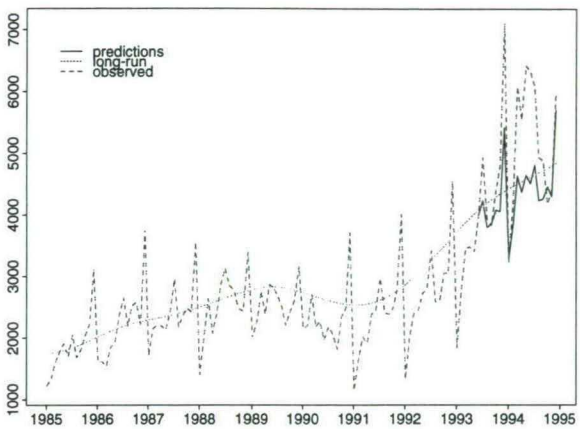

Figure 8.4: Out-of-sample predictions.

8.7. Conclusions

The main goal of this chapter was to build a model for the prediction of the production of new mortgage loans in the Netherlands, for different time horizons.

After the Dutch mortgage market was characterised and several UK studies on general mortgage demand were reviewed, we derived a theoretical reduced-form equation for the production of new mortgages, applying a simple utility maximisation principle (following Hall and Urwin [HU89]).

To formulate an empirical model, we departed from an idea of Engle *et al.* [EGH91], and combined long-run and short-run aspects of the mortgage market into a single error-correction model. The power of the error-correction model was enlarged by allowing for yet unspecified nonlinearities in the short-run part.

Neural networks were employed to search for possible short-run nonlinearities in the data generating process. The mathematical representation of a feed-forward neural network with skip-layer provides a convenient formula which can represent different specifications of error-correction models. The degree of nonlinearity (flexibility) is determined by the number of hidden units of the neural network and the penalty term. Selection of the final neural network model was performed on the basis of cross-validated goodness-of-fit measures.

The following results concerning the mortgage loan market resulted from our research. The main determinant of the long-run perspective on the production of new mortgage loans is the value of the privately owned housing stock. In the short-run, seasonality explains most of

the variation in the production of new mortgage loans. This seasonality is likely due to the consumer's house buying behaviour.

We have not found noticeable evidence of nonlinear short-run aspects in the production of new mortgage loans. Nevertheless, we are of the opinion that the proposed neural network extension of the error-correction model is useful. Hence, the specification search is partly automated, and need not be restricted to simple parametric functions. Further, the standard linear regression model is encompassed as a special case, and more complex specifications can be compared within a single framework.

Finally, the resulting model generates predictions for different horizons properly, although the quality for the mortgage production in mid-1994 is low. We conjecture that these malpredictions are caused by an unequalled high number of households changing mortgage loans (due to a sharp decrease in the mortgage interest rate in that period).

We conclude that the error-correction model that merges different models for the long-run and short-run, in which the latter is represented by a flexible form (neural network), is a promising approach for practical economic and financial forecasting.

Chapter 9

Exchange Rate Modelling

9.1. Introduction

"It is now recognized that empirical exchange rate models of the post-Bretton Woods era are characterized by parameter instability and dismal forecast performance..." [MR91]. The pessimism about the prediction quality of exchange rate models has become generally accepted after the publication of the influential paper by Meese and Rogoff [MR83]. These authors performed a large number of statistical tests, indicating that not a single economic model of exchange rates was better in predicting bilateral exchange rates during the floating-rate period than the simple random walk model, which posits that all future values of the exchange rate are equal to today's rate.

However, in [MT92]–a good survey paper on exchange rate determination – it is stated that foreign exchange rate participants focus more on fundamentals in predictions for longer horizons, and that more attention might be paid to modelling these fundamental determinants of long-term prediction.

Several approaches have been tried to improve the quality of existing structural exchange rate models. Some of these approaches have considered the incorporation of nonlinearities in the models. Diebold and Nason [DN90], for example, state that "...In summary, there appears to be strong evidence, consistent with rigorous economic theory, that important nonlinearities may be operative in exchange rate determination...". They further observe that, despite the routinely occurring statistical rejections of linearity in exchange rate models, no nonlinear model has been found in the literature (yet) that can significantly outperform even the simplest linear model in out-of-sample forecasting. Although Diebold and Nason used a powerful nonparametric prediction technique (locally-weighted regression), they were generally unable to improve upon a simple random walk in out-of-sample prediction of ten major dollar spot rates in the post-1973 period, in which the dollar exchange rates are floating. Also Meese and Rose [MR91] end up

with a negative conclusion: "...we do conclude that incorporating non-linearities into existing structural models of exchange rate determination does not at present appear to be a research strategy which is likely to improve dramatically our ability to understand how exchange rates are determined".

The exchange rate literature usually restricts the application of nonparametric approaches to locally-weighted regression techniques [MR91, MR90, DN90], which are in principle generalisations of the standard nearest neighbour technique. It is generally recognised that nonparametric modelling based on local approximations becomes difficult in high-dimensional spaces due to the increasing sparseness of the data (see Chapter 2). In macroeconomic models most data are typically sparsely distributed; data on economic fundamentals are available on a monthly basis at best, which limits the amount of data available to (say) a few hundred observations. Consequently, the principle of local averaging is likely to fail in macroeconomic modelling problems.

The foregoing does not necessarily imply that model-free regression modelling is impossible in economics. When a low-dimensional representation is embedded in the data, dimensionality reduction methods may be applied successfully. One such method is neural network regression, which we will use in this chapter. Alternatives to neural networks were discussed in Chapter 2. During the past few years there has been a noticeable increase of neural network applications in economics and finance (see Chapter 6). However, to the best of our knowledge, no studies have been performed yet that apply neural networks to structural exchange rate modelling.

This chapter, which is an extended version of Verkooijen [Ver95], examines whether introducing nonlinearities into theoretical models of exchange rate determination improves the prediction power of these models. In the empirical part neural networks are employed to investigate the nonlinearity hypothesis for the exchange rates of the Japanese yen-US dollar, the British pound-US dollar, the Deutsche mark-US dollar, and the Dutch guilder-US dollar.

More specifically, we will test whether the hypothesised fundamental determinants of the structural models that we consider, do in fact affect the exchange rate, without making auxiliary assumptions about the functional form of the relationship.

The outline of this chapter is as follows. Section 2 introduces the theoretical structural exchange rate models, which form the basis for the analyses in subsequent sections. In section 3 empirical (testable) models of exchange rate determination are formulated, based on the theoretical models of section 2. In section 4 the characteristics of the collected data are examined. Section 5 outlines the methodology for assessing predictive performance, and examines the long-run and short-run predictive power of the selected exchange rate models, specified in linear and in neural network form respectively. Section 6 concludes the chapter.

9.2. Theoretical Models of Exchange Rate Determination

There are several theories on exchange rate determination [BM89, MT92]. In many theories two general hypotheses play a prominent role, the Purchasing Power Parity (PPP) hypothesis and the Uncovered Interest rate Parity (UIP) hypothesis. The main idea of the PPP-hypothesis is that exchange rates and national consumption price indices will adjust proportionally so as to maintain a given currency's purchasing power across boundaries, which means that the real value of a given currency will be the same in all countries at any moment in time. The UIP-hypothesis states that, in equilibrium, the interest rate differential among countries must be equal to the expected rate of change of the exchange rate. In the next subsections we will make these assumptions more explicit and explain their impacts on exchange rate models.

9.2.1. The PPP-hypothesis

Consider two countries i and j, each with a bundle of n tradeable goods with average (consumer) prices P_i and P_j:

$$P_i := \sum_{k=1}^{n} \alpha_k p_{i,k} \quad \text{and} \quad P_j := \sum_{k=1}^{n} \beta_k p_{j,k},$$

where α and β denote bundle weights and $p_{i,k}$ the price of good k in country i. Define the percentage (consumer) price differential between countries i and j as:

$$dp_{ij} := \log P_i - \log P_j - \log S_{ij},$$

with S_{ij} the nominal exchange rate between i and j's currencies (expressed as units of i's currency per unit of j's currency). Then, under the PPP-hypothesis dp_{ij} is zero if, for example, the bundle weights between the two countries are identical for corresponding goods.

In practice, countries utilise different bundles of goods and price indices $P_i/P_{i,0}$, where 0 indicates the base year. Hence, the percentage (consumer) price *index* differential between countries i and j can be written as:

$$q_{ij} = \log \frac{P_i}{P_{i,0}} - \log \frac{P_j}{P_{j,0}} - \log S_{ij}. \tag{9.1}$$

To simplify our notation, we will denote $\log(P_i/P_{i,0})$ by p_i, and $\log S_{ij}$ by s_{ij}. Obviously, for any sample observation at time t, the time differentials satisfy:

$$q_{ij,t} - q_{ij,t-1} = dp_{ij,t} - dp_{ij,t-1},$$

which implies that when modelling in time differences the distinction between prices and price indices becomes irrelevant. Under the PPP-hypothesis, q_{ij} is assumed to be zero. The nominal

exchange rate satisfying this hypothesis will be denoted, henceforth, by s^*_{ij}. Notice that this s^*_{ij} will generally be different from the observed (spot) exchange rate s_{ij}, due to transportation costs, trade restrictions, speculation, and governmental stabilisation policies (see [WP95]).

9.2.2. The CIP- and UIP-hypotheses

Consider an economic agent who requires a certain amount of foreign currency, say, dollars, for use after a specific period of time, say, one month. If this economic agent is risk averse, he is expected to buy foreign currency now, provided he expects that buying at the current spot exchange rate is more favourable than buying at the one month's forward rate. This forward rate $f_{ij,t}$ is the rate agreed upon now for an exchange of currencies at an agreed specific future point in time. The consequence of buying at the current spot rate is that the foreign interest rate (instead of the domestic interest rate) is received, assuming the money is held in a foreign deposit. Since both options are riskless, it is expected that they yield the same rate of return; otherwise, arbitrage would generate riskless profits, provided that there are no barriers to arbitrage across international financial markets. The forward premium (or the opposite forward discount) at a certain maturity is the percentage difference between the current forward rate of that maturity and the current spot rate. Hence, under the Covered Interest rate Parity hypothesis (CIP-hypothesis), this interest rate differential is assumed to be equal to the forward premium (in any time period):

$$\log f_{ij} - s_{ij} = r_i - r_j, \tag{9.2}$$

where r_i denotes the nominal (short term) interest rate of country i.

When a trader expects the future spot exchange rate to be lower than the current forward rate, it may be attractive for this trader to wait until next month; thereby taking the risk of the spot rate being higher than the current forward rate. In this case actors on the forward market are prepared to pay for a risk premium, which equals the difference between the forward rate and the expected future exchange rate. If no risk premium exists in the currency market (which means that the expected future exchange rate and forward rate coincide) CIP implies the Uncovered Interest rate Parity (UIP) condition.

Under the UIP-hypothesis capital markets are assumed to be fully integrated, so that the domestic and the foreign assets are perfect substitutes and international capital is perfectly mobile. Furthermore, financial markets are assumed to be fully efficient. This assumption implies that there are no transaction costs, no differences in national tax systems on capital incomes, and no risk premia in forward markets. Then, under the UIP-hypothesis the rates of return on domestic and foreign assets (expressed in the same currency) are equal:

$$r_{i,t} = r_{j,t} + s^e_{ij,t+k} - s_{ij,t}, \tag{9.3}$$

where the superscript "e" denotes the market's expectation based on information at time t ($s^e_{ij,t+k} := E[s_{ij,t+k}|I_t]$, where I_t denotes the information available at time t), and k denotes the period of maturity. The UIP-hypothesis is the cornerstone parity condition for testing foreign exchange rate market efficiency; it assumes rational expectations and risk neutrality. In an efficient market, prices should fully reflect the information available to the market participants and it should be impossible for traders to earn excess returns due to speculation. It is important to notice that only if the nominal interest rate differential is identical to a constant and if expectations are rational, the UIP implies a random walk in the exchange rate (with drift if the constant is non-zero). In general, however, the random walk model is inconsistent with the UIP-hypothesis.

9.2.3. Monetary and Portfolio Models

Monetary models of exchange rate determination were developed after the March 1973 collapse of the (Bretton Woods) fixed exchange rate regime. These models are descendants of the Mundell-Fleming type of models (see [Mun63, Fle62]) .

Several versions of these monetary exchange rate models have been put forward, giving rise to two main types of models: the Flexible-Price Monetary Model (FPMM) due to Frenkel [Fre76] and Bilson [Bil78], and the Sticky-Price Monetary Model (SPMM) due to Dornbusch [Dor76] and Frankel [Fra79]. The modelling strategy is similar for both types: aggregated macroeconomic relationships are used to obtain a semi-reduced form equation which specifies the level of the (logarithmic) nominal exchange rate as a log-linear function of fundamental factors.

The starting point for both types of models is Cagan's money demand function for hyper-inflation (see [Cag56]) for a country: the logarithmic demands for real monetary balances are assumed to be linear functions of the logarithmic real national income and the nominal interest rate,

$$m^d = p + \alpha y - \beta r + \alpha_0, \quad (\alpha, \beta > 0), \tag{9.4}$$

with m^d the logarithm of a country's nominal money demand, p the logarithm of the price index, y the logarithm of real national income, r the nominal short term interest rate level, α the domestic income elasticity, and β the domestic interest rate semi-elasticity of the demand for money.

In the following subsections the two monetary exchange rate models will be explicitly derived and compared. Additionally, the portfolio balance model (PBM), which is non-monetary, will be discussed.

9.2.3.1. The Flexible-Price Monetary Model (FPMM)

Consider the following FPMM-assumptions:

1. prices fully adjust such that foreign and domestic commodity markets clear instantaneously;
2. there exists complete equilibrium in the domestic and foreign money markets, for any country: $m^d = m^s = m$, ;
3. national incomes are at their full-employment levels;
4. the PPP-hypothesis is continuously valid with a corresponding exchange rate s^*_{ij}.

Then the spot nominal exchange rate can be expressed by substituting Cagan's money demand function (9.4) into the PPP-hypothesis of section 9.2.1, yielding:

$$s^*_{ij} = (\alpha_{0,i} - \alpha_{0,j}) + (m_i - m_j) - \alpha_i y_i + \alpha_j y_j + \beta_i r_i - \beta_j r_j, \tag{9.5}$$

which is the fundamental flexible price monetary equation. In this equation an increase in the domestic money supply, relative to the foreign money stock, will lead to a depreciation of the domestic currency in terms of the foreign currency. A rise in domestic real income will lead to an appreciation of the domestic currency (other things equal). Similarly, a depreciation of the domestic currency follows after an increase in the domestic interest rate.

If the income elasticities on the one side and the interest rate semi-elasticities on the other side are assumed to be equal for both countries ($\alpha_i = \alpha_j$; $\beta_i = \beta_j$), equation (9.5) reduces to

$$s^*_{ij} = (\alpha_{0,i} - \alpha_{0,j}) + (m_i - m_j) - \alpha (y_i - y_j) + \beta (r_i - r_j), \tag{9.6}$$

where the logarithmic nominal exchange rates are determined as a linear combination of differences between domestic and foreign fundamentals.

A basic problem with the FPMM is that it assumes continuous PPP, so that the (logarithm of the) real exchange rate cannot vary over time, not even in the short run. This is in contrast with reality: although PPP existed during the 1920s, it largely collapsed during the recent floating rate period, which started in March 1973 (see [Fre81], [MP91]).

Therefore, we need a monetary model for nominal exchange rates with incomplete competition in the market of tradeable goods with sticky prices, at least for the short run. The Sticky-Price Monetary model (treated in the next subsection) remains fundamentally monetary, since attention remains focused on equilibrium conditions in the money market.

9.2.3.2. The Sticky-Price Monetary Model (SPMM)

The SPMM is built on the assumptions of

1. a finite adjustment speed in the commodity market with sluggish prices (sometimes leading to short-term 'overshooting' because of a slow adjustment of these commodity prices; see [Dor76]);
2. clearance of the commodity market in the long run;
3. instantaneous money and asset market equilibria with perfect substitutability of domestic and foreign non-money assets and perfect capital mobility (reflected in the UIP - hypothesis).

Following Mundell [Mun63] and Fleming [Fle62] we suppose incomplete competition in the commodity market. Then, country i's commodity demand is assumed to be dependent on real exchange rates, real national income of country j, and short term real interest rates:

$$y^d_{i,t} = \beta_{0,i} + \beta_{1,i}\,(s_{ij,t} - p_{i,t} + p_{j,t}) + \beta_{2,i}\,y_{j,t} - \beta_{3,i}\,(r_{i,t} - \pi_{i,t}), \tag{9.7}$$

where $\pi_t := p_{i,t} - p_{i,t-1}$. When country j acts as the domestic country, then i and j need to be interchanged in equations (9.8) and (9.7).

The general principle of SPMM is that prices do not adjust instantaneously: sticky prices. The price-adjustment equation is assumed to be dependent on the commodity market disequilibrium, that is,

$$\pi_{i,t} := \gamma_i\,(y^d_{i,t} - y_{i,t}) \tag{9.8}$$

with γ_i the positive price adjustment speed for country i, $y^d_{i,t}$ country i's commodity demand, and $y_{i,t}$ country i's national income. Hence, a shortage of demand will evoke decreasing prices, which, according to (9.7), will result in a rise of aggregate demand. This process will repeat itself, until the domestic commodity market is cleared; the higher the adjustment speed, the quicker the commodity market equilibrium will be reached.

The exchange rate regime is determined by the UIP-assumption. After an initial disturbance, a new equilibrium exchange rate will emerge in the long run (the 'target-exchange rate' $\bar{s}_{ij}$); in the short run the exchange rate adjustment for country i will take place at the adjustment speed θ_i:

$$s^e_{ij,t+1} - s_{ij,t} = \theta_i\,(\bar{s}_{ij} - s_{ij,t}), \quad 0 < \theta_i < 1. \tag{9.9}$$

Now consider again Cagan's money demand function (9.4), the UIP-hypothesis (9.3), and the above relationships (9.7-9.9). After substitution and definition of the equilibrium commodity price and the long run PPP-hypothesis (see equation (9.6)), we find

$$p_{i,t} = m_{i,t} - \alpha_{1,i}\,y_{i,t} + \alpha_{2,i}\,r_{j,t} + \alpha_{2,i}\theta_i\,(\bar{s}_{ij} - s_{ij,t}) - \alpha_{0,i} \tag{9.10}$$

$$\bar{s}_{ij} = (m_i - m_j) - \delta_1\,(y_i - y_j) + \delta_2\,(r_i - r_j) + \delta_0 \tag{9.11}$$

$$\begin{aligned} s_{ij,t} = {} & \frac{1}{\beta_{1,i}}[y_i - \beta_{2,i}\,y_j + \beta_{3,i}\,r_i - \beta_{3,i}\,p_{j,t} \\ & + (\frac{1}{\gamma_i} - \beta_{3,i} + \beta_{1,i})\,p_{i,t} - (\frac{1}{\gamma_i} - \beta_{3,i})\,p_{i,t-1} - \beta_{0,i}]. \end{aligned} \tag{9.12}$$

Note that the last equation (9.12) is also included with country j acting as the domestic country, that is, with i interchanged with j.

When the above equations (corresponding to the sticky price monetary model) are written into a single reduced form equation, then country i's nominal exchange rate for one unit of country j's currency satisfies

$$s_{ij,t} = f(m_{i,t}, m_{j,t}, y_{i,t}, y_{j,t}, r_{i,t}, r_{j,t}, p_{i,t}, p_{i,t-1}, p_{j,t}, p_{j,t-1}). \tag{9.13}$$

As we have already indicated, the above models are called monetary because they focus on the equilibrium conditions in the money market. They also assume perfect substitutability of domestic and foreign non-money assets so that the corresponding markets can be aggregated into a single extra market (a market of 'bonds'). This perfect substitutability assumption will be relaxed next in the Portfolio Balance Model of exchange rate determination (see [BH85]).

9.2.3.3. The Portfolio Balance Model (PBM)

The key assumption of the PBM is the imperfect substitutability between domestic and foreign assets. This model will be stock-flow consistent, in that it allows for current account imbalances to have a feedback effect on wealth and, hence, on long run equilibrium.

The net financial wealth of the private sector can be subdivided into three components: nominal domestic money M_i, domestically issued bonds B_i (which can be government debt held by the domestic private sector), and foreign bonds B_j denominated in foreign currency and held by domestic residents (which can be interpreted as net claims on foreigners held by the private sector). In a regime of floating exchange rates, a current account surplus on the balance of payments must be exactly matched by a capital account deficit, i.e., by capital outflow and, hence, by an increase in the net foreign indebtedness B_j to the domestic economy. Therefore, current account imbalances will determine exchange rate changes.

Furthermore, the assumption of imperfect substitutability of domestic and foreign assets is equivalent to the assumption of a risk premium, separating expected depreciation and the domestic-foreign interest rate differential (implying a collapse of the UIP-hypothesis). In the PBM this risk premium will be a function of relative domestic and foreign debts.

Summarising, the reduced form equation for the nominal exchange rates may be written under the PBM as:

$$S_{ij,t} = f(M_{i,t}, M_{j,t}, B_{i,t}, B_{j,t}, FB_{i,t}, FB_{j,t}), \tag{9.14}$$

where $FB_{i,t}$ and $FB_{j,t}$ denote foreign holdings of domestic and foreign bonds respectively. Taking account of the above arguments, the four last terms may be replaced by the domestic and foreign accumulated current account surplusses.

The logarithmic nominal exchange rate models (9.6), (9.13), and the logarithmic version of (9.14) may be compared, using appropriate statistical tests (Lagrange Multiplier test). There is

room left for a synthesis of the monetary and portfolio balance models, where aspects of various models should be considered simultaneously.

9.3. Empirical Models

In the previous section we introduced the three main types of structural exchange rate models and discussed the underlying hypotheses: the flexible price monetary model, the sticky-price monetary model, and the portfolio balance model. These models are often selected in the recent literature [MR83, MR91, MT92, CT95], perhaps due to their moderate data requirements.

Since we examine exchange rates against the US dollar, the notation used in the previous section can be slightly simplified: the subscripts i and j are omitted; instead all fundamentals corresponding to the U.S. carry a "*" mark.

The three models, which we will empirically test in the remainder, are subsumed in

$$s_t = f(r_t - r_t^*, m_t - m_t^*, ip_t - ip_t^*, \pi_t - \pi_t^*, TB_t, TB_t^*) + \epsilon_t \tag{9.15}$$

with s the logarithm of the bilateral spot exchange rate (for instance, DM/$); $m - m^*$ the logarithm of the relative (ratio of foreign to domestic) nominal money supply; $ip - ip^*$ the logarithm of the relative industrial production; $r - r^*$ the nominal short-term interest rate differential; $\pi - \pi^*$ the inflation rate differential; TB and TB^* the cumulated trade balances, and ϵ is the disturbance term. Theoretically, GNP is to be preferred as a proxy for real income. GNP data, however, are available on a quarterly basis, whereas industrial production data are available on a monthly basis. Therefore, following Meese and Rogoff, we use industrial production data in our experiments.

The flexible price monetary model (FPMM) includes only the first three terms, that is, $r_t - r_t^*$, $m_t - m_t^*$, and $ip_t - ip_t^*$. The sticky price monetary model (SPMM) adds the inflation rate differential $\pi_t - \pi_t^*$. The portfolio balance model (PBM) adds the cumulated domestic and foreign trade balances.

Imposing the constraint of domestic and foreign variables (except for trade balances) entering the structural models in differential form, implies that the parameters of the corresponding domestic and foreign variables are equal in absolute size, in the case of linear regression. While this parsimoniousness assumption is conventional in empirical applications, it is a potential source of misspecification. In the subsequent sections we will investigate whether this misspecification occurs.

9.4. Data Sources and Preliminary Diagnostics

In Chapter 1 we have already distinguished between stationary and nonstationary time series data. Chapter 5 was dedicated to modelling with nonstationary time series data; resulting difficulties in the application of standard statistical inference, and solutions for these difficulties. The worst consequence of modelling with nonstationary time series data is that standard statistical tests provide evidence for a supposed relationship between economic fundamentals, whereas in fact the relationship is purely spurious. Tests for cointegration have been developed to guard against making these erroneous conclusions.

Therefore, the first step in a modelling exercise incorporates the characterisation of the data. Unit root tests (introduced in Chapter 5) are normally used for this purpose. When the various time series contain a unit root, the next step is to investigate whether these nonstationary time series drift together (are cointegrated) or drift apart (are not cointegrated).

9.4.1. Data Sources

We take most of the monthly data from the OECD series (using Datastream), which include bilateral exchange rates, industrial production index, consumer price index (total), foreign trade balance, money supply (M1), short-term interest rate, and long-term interest rate. The data not available in the OECD series, are taken from the National Accounts. The data source of each variable is reported in Table C.1 in Appendix C. In Appendix C the Figures C.1 through C.4 depict the variables corresponding to each country; the monthly series range from January 1974 until July 1994.

To facilitate neural network training with weight decay (explained in Chapter 3 and 4), we rescale the data corresponding to each explanatory variable in such a way that at least 95 percent of the data lies within the [0, 1] range and the average equals 0.5 (see section 4.6). This rescaling makes the signal transferred by each input unit comparable with the outputs of internal units, which is required for weight decay to have effect.

9.4.2. Unit-roots

Chapter 5 discussed the characterisation of time series by the order of integration. To test each series for possible nonstationarity, we use ADF tests. Table 9.1 reports the characterisations suggested by these tests for the variables in differential form. Numerical outcomes of the tests are presented in Table C.2 in Appendix C.

In Table 9.1 most variables are characterised as $I(1)$, although the industrial production differential appears to be (trend) stationary, in three out of four cases. Trend stationarity is

Table 9.1: Results of unit-root tests

	Japan	*U.K.*	*Germany*	*Netherlands*
Exchange Rate	$I(1)$	$I(1)$	$I(1)$	$I(1)$
Nominal Interest Rate	$I(1)$	$I(1)$	$I(1)$	$I(1)$
Money Supply	$I(1)$	$I(1)$	$I(1)$	$I(0)$+c+t
Industrial Production	$I(1)$	$I(0)$+c+t	$I(0)$	$I(0)$
Inflation	$I(1)$	$I(0)$	$I(1)$	$I(1)$
Cumulated Trade Balances	$I(1)$	$I(1)$	$I(1)$	$I(1)$

denoted by "$I(0)$+c+t". It should be noted that discriminating between a trend stationary series and a random walk with drift, is difficult in a small sample.

9.4.3. Cointegration

The tests for unit roots suggested that most of the variables included in the models, can be assumed to be $I(1)$, although some variables seem to be (trend) stationary. Modelling with *levels* of variables that are $I(1)$ can give misleading results, as indicated in Chapter 5. The next step tests whether some linear combination of the variables is stationary. If this is the case, the variables are said to be cointegrated. Table 9.2 reports the ADF test for cointegration between the variables in the various models. The cointegrating relationship is estimated in *PcGive*[2] as the long run static solution of a dynamic autoregressive distributed lag (ADL) model. We include 6 lags[3] for each variable, a constant term, and a trend. The residuals from the static long run solution are then tested for stationarity, using ADF tests with critical values calculated from the response surface developed by MacKinnon [Mac91] for "with trend" models.

Table 9.2: Cointegration tests

model	*Japan*	*U.K.*	*Germany*	*Netherlands*	Critical Value (*alpha*=0.1)
flexible-price	2.09	2.64	2.52	3.05	4.20
sticky-price	2.32	3.57	2.38	4.29	4.50
portfolio	1.79	3.27	1.52	2.61	4.77

[2]Econometric software package developed by Hendry and his co-workers [Hen93]

[3]The number of lags was induced by capacity constraints of PcGive.

The number of lags included in the auxiliary regression of the residuals is determined by the most significant lag ($\alpha = 0.05$). No constant term was added, since it was already included in the long-run relationship.

Table 9.2 shows that in all cases the null hypothesis of no (linear) cointegration cannot be rejected. The data do not seem to confirm the three theoretical models of exchange rate determination. This conclusion is not altered when the models are estimated in unrestricted form, which incorporates foreign and domestic variables separately. In particular, the evidence for cointegration is weakened, since the number of variables is doubled. Additionally, the corresponding critical values are not available in the literature.

In Chapter 5 (section 4) we argued that if no evidence can be found for linear cointegration, then there may exist a nonlinear cointegration relationship. Therefore, next step in the cointegration analysis tests for the presence of nonlinear cointegration by neural networks. One main drawback of this approach is the huge computational effort required, especially in simulating the critical values, which depend on several neural network parameters. We have to make some concessions regarding optimality and efficiency of the test. To reduce the computational burden, we adopt the same neural network parameters for each exchange rate model and each country. In this way, we have to simulate only three critical values, namely for four, five, and seven series. We take a neural network with three hidden units. The weight decay parameter is taken to be 0.001, and the number of observations equals 246. Further, no multiple restarts are employed in the neural network training process. The residuals of the neural network versions of the flexible-price, sticky-price, and portfolio models are tested for a unit root using the neural network ADF test. The required critical values are generated as explained in section 5.4. Corresponding to the linear ADF tests for cointegration, the number of lags in the neural network ADF test is determined as the highest lag (maximum 13) that is significant at a 5% level. The results are shown in Table 9.3.

Table 9.3: Neural network ADF tests

model	*Japan*	*U.K.*	*Germany*	*Netherlands*
flexible-price	4.51	5.16	3.05	2.70
sticky-price	3.54	5.23	4.88	4.73
portfolio	4.04	4.90	3.78	5.03

note: The critical values for α=0.01, 0.05, and 0.10 are:

flexible-price:	6.02, 5.46, 5.19
sticky-price:	6.27, 5.73, 5.39
portfolio:	6.67, 6.07, 5.72

The tests for nonlinear cointegration do not reject the null hypothesis of no cointegration at reasonable significance levels. So, functional form misspecification does not seem to be an important explanation for the weak evidence for a long-run relationship among the economic fundamentals and the exchange rate.

While determining the number of lags to be included in the neural network ADF test of the residuals, we observed that the absolute value of the "t-ADF"-statistic decreases with the number of lags included. The highest values are observed for the "t-DF" statistic (i.e., no lags included). The DF-statistics would reject the null hypothesis of no cointegration for each model. However, leaving out lagged terms that are significant makes the ADF regression misspecified, which invalidates the DF test.

Nevertheless, Sephton [Sep94] performs the cointegration tests on the MARS algorithm by a DF test on the residuals, even though the sample sizes of the data employed in their applications seem too small to neglect the effects of the lagged terms in the ADF tests. Sephton's evidence for the existence of nonlinear cointegration can thus be questioned.

9.5. Predictive Performance Assessment

In this section we investigate the predictive power of the various exchange rate models, both in levels (long-run) and in changes (short-run). Our main objective is to examine whether nonlinear specification of the supposed relationship between the economic fundamentals and the exchange rate gives better predictive performance than the benchmark random walk model and linear specifications do.

As we have already indicated in Chapter 4, neural networks have the danger of overfitting the data. To prevent such overfitting, we employ neural network training with a weight decay term added to the least squares error function (see Chapter 3, formula (3.9)). The effect of weight decay is that large weights are penalised. Varying the weight decay parameter from low to high transforms the approximating function from highly flexible to rigid. There exists a value for the weight decay parameter that restricts the network weights such that the approximating function closely resembles the linear model estimated by OLS; a further increase of the value makes the approximating function resemble 'penalised OLS' (also known as ridge regression). So, by the weight decay parameter we determine the level of flexibility.

Cross-validation was introduced in Chapters 2 and 4 as a procedure for selecting the value of the weight decay parameter. The weight decay value suggested by cross-validation and the corresponding cross-validation MSE, immediately indicate whether (strong) nonlinearities are present in the data, or whether the OLS estimates of the parameters in the linear model have to be shrinked. In the following two subsections we shall sometimes use this information to skip the neural network results, when cross-validation indicates that no flexibility is needed. In some

cases we shall deliberately choose a weight decay parameter smaller than the one suggested by cross-validation, to enforce differences in performance between the linear models and the neural network models; of course, risking bad predictions due to overfitting.

9.5.1. Methodology for Out-of-sample Model Comparison

In line with Meese and Rogoff [MR83], we will compare the models using rolling regressions (also called recursive estimation or running regression). This means that we start with an initial estimation of the model, using (say) the first n_0 observations. We then make predictions for the remaining part of our sample (n-n_0); after this we include the next observation in the parameter estimation set (which now consists of $n_0 + 1$ observations), and again predict the the response variables in the remaining set of observations. This procedure is repeated until the training set equals the total sample. In this way we have constructed a set of $(n - n_0)$ 1-step ahead predictions, $(n - n_0 - 1)$ 2-steps ahead predictions, or in general, $n_k = (n - n_0 - k + 1)$ k-steps ahead predictions ($k < n - n_0$).

Note that the structural models require forecasts of their predictor variables in order to generate predictions of the exchange rate. In line with what is usually done in the literature in this case, we use the actually realised values of predictor variables. Consequently, the results are optimistically biased.

As our principal criterion for comparison we take RMSE

$$RMSE(k) = \left\{ \sum_{p=n_0}^{n-k} [\hat{y}_{p+k} - y_{p+k}]^2/(n-k) \right\}^{1/2}, \tag{9.16}$$

where k denotes the prediction horizon (in months), y_{p+k} the observed value of the response variable at time $p + k$, and $\hat{y}_{p+k}$ the response value estimated by a model with parameters estimated from the data set $\{(\mathbf{x}_i, y_i)\}_1^p$.

9.5.2. Long-Run Predictions

The cointegration analyses indicate that if there is a relationship between the exchange rate and the selected economic fundamentals, it is tenuous at best. In this section we examine whether –despite the weak evidence for cointegration– the exchange rate models can tell more about the future than the random walk model ($\hat{s}_{t+k} = s_t$, $k = 1, 2, \ldots$) does.

When the models estimated in the levels of the variables are in fact spurious, the out-of-sample prediction will show no improvement over the prediction accuracy of the random walk model. We regard the examination of the predictive accuracy of the models (in levels) as complementary to the cointegration test, which tests for the existence of the supposed (equilibrium)

relationships between the economic fundamentals and the exchange rate. Theoretically, it is possible that a cointegration relationship escapes the Engle-Granger cointegration test (applied in section 9.4.3). Hence, the assumptions of the cointegration test were not all satisfied; some of the variables were $I(0)$ rather than $I(1)$. Furthermore, Mark [Mar95] finds evidence for long-horizon predictability (in levels) of the exchange rate by some economic fundamentals.

Table 9.4: Accuracy of long-run predictions: RMSE(OLS)/RMSE(neural network)

model	1 month	6 months	12 months	18 months	24 months
			Japan		
flexible-price	0.23/0.09	0.26/0.15	0.30/0.22	0.33/0.28	0.35/0.34
sticky-price	0.23/0.10	0.27/0.19	0.31/0.25	0.34/0.32	0.38/0.38
portfolio	0.23/0.09	0.27/0.17	0.31/0.24	0.35/0.33	0.38/0.38
random walk	0.03/0.03	0.08/0.08	0.11/0.11	0.14/0.14	0.17/0.17
			United Kingdom		
flexible-price	0.21/0.11	0.24/0.15	0.28/0.17	0.32/0.19	0.35/0.24
sticky-price	0.20/0.15	0.23/0.13	0.26/0.17	0.29/0.20	0.31/0.21
portfolio	0.19/0.12	0.23/0.15	0.26/0.16	0.31/0.17	0.33/0.21
random walk	0.04/0.04	0.13/0.13	0.17/0.17	0.19/0.19	0.20/0.20
			Germany		
flexible-price	0.30/0.25	0.35/0.38	0.39/0.49	0.41/0.56	0.41/0.57
sticky-price	0.29/0.18	0.35/0.27	0.41/0.36	0.44/0.41	0.44/0.44
portfolio	0.27/0.18	0.33/0.27	0.38/0.36	0.40/0.43	0.41/0.45
random walk	0.04/0.04	0.12/0.12	0.16/0.16	0.20/0.20	0.20/0.20
			The Netherlands		
flexible-price	0.21/0.21	0.25/0.28	0.28/0.34	0.30/0.38	0.32/0.41
sticky-price	0.17/0.16	0.20/0.24	0.23/0.34	0.24/0.40	0.24/0.43
portfolio	0.20/0.13	0.23/0.19	0.27/0.23	0.29/0.27	0.29/0.29
random walk	0.04/0.04	0.11/0.11	0.15/0.15	0.19/0.19	0.19/0.19

To examine the possible existence of a long-run relationship, both linear and neural network exchange rate models in levels are employed. The models are compared on the RMSE criterion, described in the previous section. The prediction performance of the random walk model is included as a benchmark in the comparison; random walk models are often used for this purpose in the literature.

The following procedure is followed to construct Table 9.4. The initial linear and neural network models are estimated on the first 140 observations, including the determination of the weight decay value for the neural networks. The number of hidden units was fixed at four. Five restarts are used to find a neural network representation of a particular exchange rate model. The cross-validation procedure, which is employed to select the weight decay value, suggests

Table 9.5: Long-run predictions with unrestricted models RMSE(OLS)

model	1 month	6 months	12 months	18 months	24 months
			Japan		
flexible-price	0.13	0.16	0.20	0.23	0.26
sticky-price	0.10	0.13	0.14	0.17	0.18
portfolio	0.10	0.13	0.14	0.16	0.18
			United Kingdom		
flexible-price	0.25	0.30	0.35	0.40	0.44
sticky-price	0.23	0.28	0.33	0.38	0.42
portfolio	0.20	0.24	0.27	0.32	0.35
			Germany		
flexible-price	0.23	0.30	0.35	0.35	0.34
sticky-price	0.22	0.28	0.33	0.34	0.34
portfolio	0.24	0.27	0.32	0.33	0.32
			The Netherlands		
flexible-price	0.21	0.26	0.30	0.32	0.33
sticky-price	0.20	0.24	0.28	0.29	0.31
portfolio	0.19	0.22	0.24	0.25	0.24

a value between 0.01 and 0.001; the corresponding cross-validation error is smaller than the cross-validation error of a linear model estimated by OLS. The initial model is then used to predict the remaining part of the data. Then, the next observation is added to the estimation set, and the parameters (weights) are updated, using the latest values to depart from. This procedure is repeated until all observations are in the estimation set. Finally, all one-month-ahead predictions are collected, and the corresponding RMSE is calculated; the results are shown in the first column in Table 9.4. The same is done for 6, 12, 18, and 24 months-ahead predictions; the corresponding results are shown in the next columns of the table.

We make two conclusions from Table 9.4. First, no structural exchange rate model –linear or neural network– achieves better predictions than the random walk model for prediction horizons up to two years ahead. It should be recalled that actual values were inserted for the independent variables, which makes the results even less promising. The results, however, are in line with the findings of other studies [MR83, MR91, DN90], and support the very weak evidence we found for linear and nonlinear cointegration. Second, the neural networks outperforms the linear models in most cases. However, with a large prediction horizon (18 and 24 months) the neural network's predictions are worse than the linear model's predictions, in general. This may be due to extrapolation difficulties, which seems to hurt neural networks more than the linear models.

We also investigate the out-of-sample prediction capacity of unrestricted models, i.e., foreign and domestic variables are included separately. The results are shown in Table 9.5. Since the

neural network predictions closely approximate the predictions of the linear model, we left the former out. The most striking observation is that the predictions for the Japanese Yen against the US dollar exchange rate have been improved considerably. The random walk models could again not be beaten by the structural models (either a linear model or a neural network). Despite these disappointing results, we make some observations on modelling for prediction that seem worth mentioning.

When the models are specified in unrestricted form, the number of variables is doubled, which increases the variance of the OLS-parameter estimates. This may lead to bad long-run predictions. The neural networks weights are determined by minimising the compound loss function, consisting of the sum of squared errors and the sum of squared weights (weight decay learning). The effect of weight decay is a reduction of the variance of the weights, at the expense of a (somewhat) higher bias. Weight decay is particularly effective in the case of many connections and relatively few observations. When biased estimation is applied to the linear model by adding the same penalty term to the error function, the long-run predictions may improve as well. In case the linear unrestricted flexible-price model for the UK is estimated by penalised OLS with a weight decay value of 10, the corresponding row in Table 9.5 becomes

flexible-price 0.18 0.19 0.22 0.24 0.26.

So, the prediction performance increases significantly. Despite the positive impact that regularisation has on the predictions, the performance of the random walk model is still out of range.

Another observation concerns the chance of drawing faulty conclusions from the one-period-ahead prediction criterion, when that criterion is used for discerning between the predictive power of neural networks (or flexible regression methods in general), and the predictive power of linear models, in the case of $I(1)$ variables. In this case it pays off to overfit the observations in the training set, presuming that the performance assessment is done on one-period-ahead prediction errors. To illustrate this statement, we fitted a redundant neural network (eight hidden units and weight decay value λ=0.0001) to the model for the Yen-Dollar exchange rate, including ip, m, ip^*, r^*, and m^*. These particular variables were selected on the basis of their sluggishness in changing. The resulting one-period-ahead RMSE was 0.06, which is clearly the best among the structural models; see the first part of Table 9.4 and Table 9.5. The 12, 24, and 36-periods ahead prediction RMSE, however, dramatically increased to 0.49, 0.94, and 1.56, respectively. Compared to the values of the corresponding rows in Table 9.4 and Table 9.5, these values are excessively high.

An intuitive explanation of the foregoing is as follows. Assume the series of interest y_t is generated by

$$y_t = y_{t-1} + \nu_t \quad \nu_t \sim \text{i.i.d.}(0, \sigma^2).$$

Let the series $\mathbf{x}_t$ be assumed to be useful for predicting y_t, but in reality they are *not*. Assume $\mathbf{x}_t$ to be generated by

$$\mathbf{x}_t = \mathbf{x}_{t-1} + E_t \quad E_t \sim \text{i.i.d.}(0, \sigma^2 I).$$

The hypothesised relationship f between y and $\mathbf{x}$ is determined on the data set $\{(y_i, \mathbf{x}_i)\}_{i=1}^{t}$ by a very flexible method on the one hand, and a linear model on the other hand. The flexible method will be able to approximate the last observation closely; this implies $f(\mathbf{x}_t) \approx y_t$. The linear model will in general give values less close to individual observations. Using the flexible function f to predict y_{t+1} given $\mathbf{x}_{t+1}$ then results in

$$\hat{y}_{t+1} = f(\mathbf{x}_{t+1}) = f(\mathbf{x}_t + E_{t+1}) \approx f(\mathbf{x}_t) \approx y_t,$$

assuming E_t is sufficiently small. This implies that the one-step-ahead predictive performance of f will be close to the predictive performance of the random walk model. The linear model, which has a larger bias, will fit the data less precisely. So, in that case $f(\mathbf{x}_t) \approx y_t$ will not hold, making the linear model's predictive performance worse than that of the flexible model. (Hence, y_t is the best predictor of y_{t+1} by construction). The investigator should not conclude that y_t is nonlinearly related to $\mathbf{x}_t$ by f, argueing that combining $\mathbf{x}_t$ nonlinearly yielded better one-step-ahead predictions than combining them linearly.

The spurious relationship is revealed when the prediction horizon is enlarged. In case a real fundamental relationship were found, the performance would not decrease so much. However, if the relationship is spurious, the performance will decrease rapidly when the prediction horizon is enlarged.

9.5.3. Short-Run Predictions

The previous sections showned no evidence for the presence of a long-run relationship between the exchange rate and the economic fundamentals proposed by theories on exchange rate determination. In this section we examine whether short-run predictions can be made from the various exchange rate models, including the variables in first-differenced form.

The application of standard econometric inference to a dynamic (ADL) form of the exchange rate models with first-differenced variables reveals a strong significance of the exchange rate change in the previous period for all four countries. The details on this analysis are presented in Appendix C section C.3. This provides evidence against the simplest theory of "no change" in the level of the exchange rates. Therefore, we will also consider the univariate model

$$\Delta s_t = f(\Delta s_{t-1}, \ldots, \Delta s_{t-k})$$

for the exchange rate changes.

Table 9.6: Short-Run Predictions RMSE

country	parsimonious (OLS/NN)	complete (OLS/NN)	univariate (OLS/NN)	random walk
Japan	0.221/0.221	0.230/0.214	0.217/0.216	0.223
UK	0.289/0.291	0.305/0.303	0.290/0.290	0.323
Germany	0.247/0.224	0.269/0.254	0.243/0.273	0.264
Netherlands	0.250/0.250	0.266/0.259	0.248/0.248	0.269

Table 9.6 presents the RMSEs of one-period-ahead predictions made by the parsimonious models displayed in Table C.3 (Appendix C), the complete (portfolio) models with two lags for each variable, and the univariate time series model, all estimated by OLS and by a neural network; additionally, Table 9.6 gives results for the random walk model $\Delta s_t = \epsilon_t$ where ϵ_t is i.i.d $(0,\sigma^2)$. The initial models in the recursive estimation procedure are estimated on the first 180 observations. The neural network versions of the parsimonious models are estimated with two hidden units and a weight decay value of 0.1. The neural network versions of the complete models are constructed with two hidden units and a weight decay value of 5. The neural network parameters have been determined by cross-validation, and indicate that if nonlinearities are present, the effects are tenuous. Hence, the small number of hidden units and the relatively large value of the weight decay parameter suggested by cross-validation are attempts to reduce overfitting (rather than attempts to explore nonlinearities). Table 9.6 indicates that some short-run prediction is possible; the RMSEs of the one-period-ahead predictions are smaller than the RMSEs of the 'no-change' random walk models. In two cases (Germany and Japan) the neural network model provided somewhat better results than the corresponding linear models estimated by OLS.

The most relevant regularity that has been found in the data is that the next exchange rate will move in the same direction as it has moved in the previous period, and that the size of the change is damped by a factor of approximately 0.4 (see Table C.3).

The rolling prediction experiment, which gave rise to Table 9.6 revealed that over the last 62 months some structure is present in the exchange rate movements. However, the factor that seems most important is the change in the exchange rate from the previous month. To assess the possible impact of the economic fundamentals that were selected in the parsimonious models for the complete period 1974-1994, we perform an additional cross-validation test on the linear models, as follows. Two years of observations are repeatedly left out from model estimation, and are then predicted from the resulting model. The resulting out-of-sample predictions are compared with the actual values. The results are shown in Table 9.7. Table 9.7 shows which part of the variance in Δs_t is explained by the parsimonious linear models in Table C.3 and the

Table 9.7: Out-of-sample explained variance: R^2

country	parsimonious	univariate
Japan	0.18	0.11
UK	0.26	0.15
Germany	0.22	0.14
Netherlands	0.20	0.11

univariate model $\Delta s_{t+1} = \alpha_0 + \alpha_1 \Delta s_t$, respectively. We conclude that over the period 1974-1994 some of the selected economic fundamentals helped explain part of the variance in the exchange rate changes. Over the last 5 years, however, their effect on predictive performance was very small.

As said before, the parsimoniousness assumption may introduce a misspecification into the exchange rate models. We have examined whether this is the case for the models in first differenced form. Incorporating each domestic and foreign variable separately, as well as two lags of each, has a negligible impact on the predictive quality of these models.

9.6. Conclusions

We applied neural network specification and linear specification to three structural exchange rate models (flexible price and sticky price monetary models), and compared their out-of-sample predictive qualities. We conclude the following.

First, no evidence was found that confirms the existence of a long-run relationship (linear or nonlinear) among the exchange rate and the economic fundamentals included in the flexible price, sticky-price, and portfolio models. When the foreign and domestic variables were included as separate explanatory variables, the conclusion did not change. Consequently, long-run predictions obtained from these models were worse than predictions obtained from the 'no-change' model.

Second, when the models were estimated in first differenced form, we found some evidence of a weak structure underlying monthly exchange rate changes. The two main determinants are the previous month's exchange rate change and the change in the interest rate differential between two countries. The 'no-change' model, which implies that changes in exchange rates are random and can therefore not be predicted, is outperformed by linear models for all four countries (Germany, Japan, United Kingdom, the Netherlands). A neural network exploration for possible nonlinearities in the short-run models did not show evidence of such nonlinearities. When the foreign and domestic variables were included separately, this finding did not change.

Third, in general, biased estimation improves the predictive quality of the various models, especially for the long-run. The neural network experiments revealed that weight decay favourably affects the prediction quality of the neural network models. In some cases the neural network showed better prediction performance than the corresponding linear model estimated by OLS. In those cases, it was the regularisation by weight decay rather than the introduction of nonlinearities that was responsible. Hence, biased estimation also improved the predictive quality of the linear models considerably, especially when modelling with (nearly) collinear independent variables or with a high number of independent variables and a relatively small set of observations.

In this study neural networks were used to investigate the hypothesis that introducing nonlinearities into existing structural models of exchange rate determination improves the predictive quality of these models. Exchange rate determination has always been a difficult problem [MR83, MR91, MT92] that is characterised by very weak underlying relationships. These relationships are hard to quantify for any regression method, including neural networks. Introducing nonlinearities into current exchange rate models does not seem to be a future research direction with high expected payoffs.

Chapter 10

Summary and Conclusions

In economic data modelling one tries to find relationships among economic entities such that the data sample at hand is approximated as well as possible and that new observations will be predicted accurately. The increasing availability of computer power has stimulated research in data modelling techniques that search for an approximating function over some large classes of functions using the data sample at hand. The neural network is a popular flexible regression technique. In economics, however, most modelling is still performed using parametric methodology.

The topic of this thesis is the application of neural networks to economic and financial problems of prediction. The aim is to investigate the usability and the practical relevance of neural networks in the specification of economic (time series) models and their position among alternative (statistical) techniques. An additional aim is to stimulate cross-fertilization between the neural network field on the one hand, and the statistics and econometrics field on the other hand.

Our type of research can be characterised as *exploratory*, since we have examined (among others) the potentials of a new methodology –neural networks– for economics and finance. The global outline of the study is as follows. Part I (Chapters 1-5) discusses the theoretical aspects of economic modelling and neural networks. Chapter 1 describes the general economic modelling problem and the parametric approach to model building, which is generally accepted in econometrics. As alternatives to this parametric approach, Chapter 2 introduces several flexible regression methodologies; among them are neural networks. Different aspects of the neural network methodology are then discussed in the chapters 3 and 4. Chapter 5 discusses the usefulness of neural networks in modelling nonstationary time series. Part II (Chapters 6-10) deals with the practical aspects of applying neural networks to problems in economics and finance. Chapter 6 reviews the literature on neural network applications in economics and finance. The practical usability of neural networks is examined in three case studies, presented

in Chapters 7, 8, and 9 respectively. Chapter 10 gives the conclusions. In more detail these chapters may be summarised as follows.

Chapter 1 describes the general economic modelling process, which is divided into three parts: model specification, estimation, and evaluation. In general, these models are linear in their parameters (not necessarily in their explanatory variables). The issues that were discussed, concern both cross-section and stationary time series data. The central issue in model specification is building models without violating the standard statistical methods of inference. The methodology of model specification was reviewed from the viewpoint of the econometrics literature. In addition, (parametric) model estimation and evaluation were described.

Chapter 2 reviews several flexible regression methodologies found in the statistics literature. These methodologies search –beyond the space of parametric (linear) functions– for a suitable data approximating function, using the data at hand. The neural network is considered to be a member of this class of flexible regression methodologies. A vital issue that affects the whole class, is the bias/variance dilemma. This dilemma implies that an accurate within-sample fit goes together with a bad out-of-sample prediction performance in practice. A general characteristic of flexible regression techniques is the presence of one or more so-called flexibility parameters. These parameters determine the degree of flexibility of the resulting approximating function. The bias/variance dilemma means that better out-of-sample predictive accuracy may be obtained by reducing the flexibility of the approximating function. The choice of the flexibility parameter(s) is usually made on the basis of some measure of generalisation ability, such as the squared prediction error. Small samples make it difficult to estimate the prediction error reliably. Cross-validation is a method designed to provide an estimate in such situations.

Chapter 3 starts with the graphical and mathematical representation of neural networks. Neural network learning was discussed from a statistical perspective. It was shown that neural network learning becomes conceptually very close to statistical nonlinear regression, once the neural network architecture has been fixed. The greatest concern in applying neural networks is generalisation. Weight decay is a regularisation method which generally improves the generalisation ability of neural networks. It amounts to adding a penalty term to the standard (squared error) loss function. A common procedure for evaluating the neural network's performance is to compare its performance with the performances of alternative techniques. When more than two methods are compared, it is vital to take the multiplicity effect into account when drawing conclusions. We discussed the required statistical theory and proposed some statistical multiple comparison procedures.

Chapter 4 addresses the most important practical aspects of neural network design, namely the specification of the neural network's architecture, its components, the learning procedure, and the software package used. Simulation experiments illustrated the occurrence of many different local minima, the effectiveness of weight decay in reducing overfitting and on the number of

local minima. In order to create a transparent methodology, we formulated an explicit neural network construction procedure. This procedure comprises a principled selection of the number of hidden units and the weight decay parameter.

Chapter 5 deals with the modelling of nonstationary time series through neural networks. It starts with an intuitive illustration of the difficulties that nonstationary time series involve when prediction is the goal. Cointegration and error-correction are econometric concepts, which are designed to enable sound modelling with nonstationary time series. A first step was taken towards nonlinear generalisation of these concepts, as follows. Neural networks were used to make nonlinear cointegration and nonlinear error-correction models operational. Critical values were generated for a neural network augmented Dickey-Fuller test, which tests for the presence of nonlinear cointegration. Further, the standard linear error-correction model (ECM) was extended in a nonlinear way by the addition of a parameterised neural network to the short-run part of the ECM.

Chapter 6 forms the transition from the theoretical discussion of neural networks to their application to actual cases. The chapter's aim is to characterise the types of financial problems that neural networks have been applied to. This characterisation was done by a review of a sample of applications, drawn from the literature on neural networks applied to problems in economics and finance. We observed that neural networks are most often applied to classification tasks, such as bondrating and credit scoring. In many studies, especially in the earlier ones, the methodology was obscure, and conclusions were based on (very) small data sets. Hence, the claims were exaggerated sometimes, and should be interpreted with some caution.

Chapter 7 applies neural networks to the modelling of hedonic house prices in Boston. In this case study the data are of the cross-sectional type, the sample size is relatively large (506), and the number of explanatory variables is intermediate (13). Two parametric models and a neural network were estimated on the data in the specification set (400 observations). The predictive performances –measured by the evaluation set (106 observations)– were statistically compared by pair-wise t-tests with resampling and adjusted for the multiplicity effect. According to these tests, the neural network model achieved a predictive accuracy that is significantly better than the predictive accuracy of the two parameteric models. Since the neural network has found a better solution, the final solution deserves more analysis. To this end we proposed three measures. These measures indicate the average influence, the average absolute influence, and the degree of monotonicity in the partial relationship for each input factor.

Chapter 8 reports on a study in which neural networks were applied to the prediction of the production of new mortgage loans in the Netherlands. The model should provide predictions for horizons ranging from 1 month to 18 months. The relevant economic entities were characterised as nonstationary time series, and were measured at different time frequencies (monthly and annual). We employed an error-correction model which synthesises long-run and short-run

aspects. The long-run component was specified by a linear model based on 30 annual observations. The neural network procedure was used to explore complex (nonlinear) specifications of the short-run part of the ECM. In the Dutch mortgage case a nonlinear specification of the ECM did hardly improve the prediction quality of a linear ECM.

In Chapter 9 neural networks were used to investigate whether the introduction of nonlinearities into models of exchange rate determination improves the prediction performance (in levels) of these models. We examined three structural exchange rate models for four foreign exchange rates: the Dutch guilder/US dollar, the Japanese yen/US dollar, the Deutsche mark/US dollar, and the British pound/US dollar. As a starting point we took three well known theoretical models of exchange rate determination: the flexible price monetary model, the sticky price monetary model, and a portfolio balance type of model. Next empirical models were specified by a linear functional form and a neural network based, flexible, functional form. The long-run and short-run predictive qualities of both types of model specifications were investigated and compared to the predictive quality of a simple random walk model. The main conclusion is that including nonlinearities into the structural models of exchange rate determination barely improves their predictive quality; random walk models could not be outperformed for prediction horizons upto two years. When predicting the one month's change in the exchange rate, the pure random walk model was outperformed by a simple linear model in which the previous month's change in the exchange rate was most important. The introduction of nonlinearities into the short-run models also barely improved their predictive quality.

In Appendix A we outlined neural network learning and prediction from the perspective of Bayesian statistics. In Bayesian terminology, the weight decay term, which was described in Chapter 3, can be interpreted as a prior distribution of the weights.

In Appendix B we presented the data sources of the variables used in Chapter 8.

In Appendix C we presented the data sources of the variables used in Chapter 9, the detailed test results of unit root tests, and the results of the econometric analyses of the short-run models.

The following conclusions are drawn from the research undertaken in part I.

- In economic modelling one should follow a sound strategy when specifying a model, in order to avoid questionable models due to unbridled data mining or specification searches (Chapter 1).
- Neural networks are a member of the class of flexible regression functions and they suffer from the difficulties inherent to that class, such as the bias/variance dilemma (Chapter 2).
- Once the architecture of a neural network is fixed, neural network learning becomes conceptually very close to statistical nonlinear regression (Chapter 3).
- The innovative aspect of neural networks seems to bethe particular form of the approximating functions, not the specific learning strategies or application methodologies

(Chapter 3).

- A crucial issue when making statistically sound comparisons of predictive performances of several data modelling techniques is the multiplicity effect (Chapter 3).
- Controlling the degree of overfitting is vital for a successful application of neural networks (Chapters 3, 4).
- Weight decay is an effective method for constraining the flexibility of the resulting neural network solution (Chapter 4).
- Cross-validation, although time consuming, is of great help in selecting appropriate values for the neural network parameters, such as the number of hidden units and the weight decay value (Chapter 4).
- It is questionable whether a nonlinear cointegrating relationship constructed by a flexible regression technique is practically useful for extrapolation or for long-run predictions (Chapter 5).
- The detection of a nonlinear cointegrating relationship (in-sample) may help to improve parametric models and existing theories (Chapter 5).
- The nonlinear generalisation of the short-run part of the linear error-correction model by a neural network seems promising for practice (Chapter 5).

The following conclusions are drawn from the research undertaken in part II.

- Neural network applications solving economic and financial problems are often poorly described, so their methodology remains obscure (Chapter 6).
- Neural networks can automatically find a good specification of a regression equation with cross-sectional data. This is especially helpful when economic theory fails to suggest a suitable functional form (Chapter 7).
- Since hedonic price models can be constructed not only for houses but also for other goods than houses, neural networks certainly have potential for this particular area in economics (Chapter 7).
- Qualitative economic modelling and data collection is much more time consuming than the quantitative specification and estimation of the empirical model (Chapter 8).
- The (nonlinear) error-correction model is well suited to synthesise the short- and long-run aspects of a data generating mechanism (Chapter 8).
- The introduction of nonlinearities –by neural networks– into structural models of exchange rate determination does not improve the prediction performance considerably (Chapter 9).

Our final conclusion may be stated as follows.

Neural networks can be conveniently applied to various economic modelling problems. These neural networks can be embedded into the methodology for performing

empirical studies that is generally accepted by applied economists and econometricians. We have developed a neural network methodology that automatically indicates whether nonlinear approximations to the data are justified. Our method ensures that functional form misspecification is not a likely cause for possible unsatisfactory model performance. So, neural networks are a useful extension to the econometrician's toolbox, but they do not replace established econometric modelling and inference techniques.

One of the results of solving certain problems is that one ends up with new questions. Therefore, we suggest some directions for future research.

The practical relevance and performance of the predictive approach to neural network learning (explained in Appendix A) needs to be examined for economic and financial modelling problems in particular. It is known that in small samples the predictive approach deviates from the plug-in approach, on which this thesis elaborated. The predictive approach deals with overfitting and multiple local minima in a natural way, by integrating over the posterior weight distribution.

The understanding of cross-validation in nonlinear problems is still incomplete, and needs to be investigated by carefully designed simulation studies and by theoretical statistical studies. Especially interesting are studies that investigate the variance of cross-validation, which is influenced by several problem characteristics, such as sample size and signal-to-noise ratio, and several different neural network characteristics, such as number of hidden units, weight decay value, and network type. It is still questionable whether cross-validation is the best measure of generalisation ability to use in neural network modelling. To this end, cross-validation should be compared to other global measures of generalisation ability.

Neural network practitioners strongly need an accepted methodology. The "standardisation" of such a methodology makes the results of their empirical studies better to interpret. Statisticians could apply their expertise to develop such a standardised approach.

Most of the neural network software packages support only neural network learning that minimises the squared error loss function or Kullback-Leibler distance (section 4.5.3). Neural network learning algorithms should be developed which accept user-defined loss functions. Such learning algorithms are especially valuable when the objective of modelling is to maximise the *number* of correctly classified objects, the money profits in a financial trading situation, and the like.

In this study we have only investigated the usability of neural networks for single equation systems. In (macro)economics, however, simultaneous equations systems often are a better means to describe the underlying system. Each equation in such a model has a simple parametric form, usually linear. It is interesting to explore nonlinear (flexible) generalisations of such simultaneous equations systems, using neural networks. In such a system multiple output units

are needed to represent the various endogeneous variables.

Finally, the first step we have made towards nonlinear generalisation of cointegration and error-correction requires further theoretical and practical investigation.

Appendix A

A Bayesian View on Neural Network Learning

In this Appendix network learning with weight decay is discussed from a Bayesian perspective; also see section 3.4. Additionally, the Bayesian approach to *prediction* is outlined, in which uncertainty in the weight vector is explicitly taken into account. For a good introduction to the fundamentals of Bayesian analysis we refer to [Ber85]

A.1. Weight decay

Assume we have a data set consisting of n observations $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, where $\mathbf{x}$ denotes a vector of input variables and y the scalar output variable. The data are assumed to be independently drawn from a distribution $p(\mathbf{x}, y)$; they are normalised to mean zero and variance one.

A parameterised neural network f with weight vector $\mathbf{w}$ defines a mapping from an input vector $\mathbf{x}$ to a predicted output $\hat{y}$, namely $\hat{y} = f(\mathbf{x}, \mathbf{w})$. We model y as a function of $\mathbf{x}$, $y = f(\mathbf{x}, \mathbf{w}) + \epsilon$, assuming the noise ϵ to be Gaussian i.i.d. with zero mean. Using the Gaussian error model, the (sample) likelihood of y given $\mathbf{x}$ and $\mathbf{w}$ is given by

$$P(y \mid \mathbf{x}, \mathbf{w}) = (2\pi\sigma_o^2)^{-\frac{1}{2}} \exp\left(-\frac{(y - f(\mathbf{x}, \mathbf{w}))^2}{2\sigma_o^2}\right) \tag{A.1}$$

where σ_o is the level of inherent noise in the outputs (i.e. $\sigma_o^2 = \mathrm{Var}(\epsilon)$). The intuitive reason for the name "likelihood function" is that a weight vector $\mathbf{w}$ for which $P(y \mid \mathbf{x}, \mathbf{w})$ is large, is more "likely" to be the true $\mathbf{w}$ than a weight vector $\mathbf{w}$ for which $P(y \mid \mathbf{x}, \mathbf{w})$ is small, in that y would be a more plausible occurrence if $P(y \mid \mathbf{x}, \mathbf{w})$ were large.

In the conventional maximum likelihood approach to neural network training the negative logarithm of the sample likelihood $L(\mathbf{w})$, given by

$$L(\mathbf{w}) = \sum_{i=1}^{n} \log P(y_i \mid \mathbf{x}_i, \mathbf{w}) = \sum_{i=1}^{n} \frac{(y_i - f(\mathbf{x}_i, \mathbf{w}))^2}{2\sigma_o^2} + C \tag{A.2}$$

is minimised; the constant C does not depend on $\mathbf{w}$. When the optimal weight vector $\mathbf{w}^*$ is used in the prediction of the output y_{n+1} for a test case $\mathbf{x}_{n+1}$, this is referred to as the *plug-in* approach to prediction. The uncertainty in this prediction due to the inherent noise in the output data is given by the sample estimate of σ_o; however, the uncertainty in the estimation of $\mathbf{w}^*$ is usually not accounted for. We shall come back to this point later.

So far we have described the likelihood of classical statistics. The genuinely Bayesian features are now introduced. The main distinction betweem Bayesian statistics and classical statistics is that Bayesians combine prior information with information extracted from the data. The neural network learning method is derived from applying the simple Bayesian principle

$$\text{posterior} \propto \text{prior} \cdot \text{sample-likelihood}$$

to the training problem, where $\propto$ means "is proportional to". In Bayesian terminology the term "probabilities" corresponds to a relative measure of belief in the many possible network weight vectors. There is much literature about Bayesian analysis, priors, etc.; see [Ber85], for instance.

The prior distribution of the weights may be assumed to be Gaussian

$$P(\mathbf{w}) = (2\pi\sigma_w^2)^{-\frac{N}{2}} \exp\left(-\frac{|\mathbf{w}|^2}{2\sigma_w^2}\right) \tag{A.3}$$

where σ_w is the expected standard deviation of the weights, and N is the total number of weights. This is only one of the many priors that could be chosen (see [BW91]), and seems justified if we assume the regression to be reasonably smooth. The Gaussian prior is based on the experience that in smooth regressions positive and negative weights are encountered equally frequently, that smaller weights are more frequent than larger ones (in absolute size), and that very large weights are very unlikely. On the one hand very large weights result in networks that describe very nonlinear behaviour; on the other hand, very small weights provide neural networks with an almost linear behaviour. This is caused by the specific shape of the logistic squashing functions, which consist of an almost linear part around their center.

Using the well known Bayes' rule

$$P(A|B) = P(B|A)P(A)/P(B),$$

the posterior probabilities of the weights are obtained by combining the prior distribution of the weights with the sample likelihood:

$$P(\mathbf{w}|(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)) = \frac{P(\mathbf{w})P((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)|\mathbf{w})}{P((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n))} \tag{A.4}$$

$$= \frac{P(\mathbf{w})P(y_1,\ldots,y_n|\mathbf{x}_1,\ldots,\mathbf{x}_n,\mathbf{w})}{P(y_1,\ldots,y_n|\mathbf{x}_1,\ldots,\mathbf{x}_n)} \quad (A.5)$$

$$= \frac{P(\mathbf{w})\prod_{i=1}^{n} P(y_i|\mathbf{x}_i,\mathbf{w})}{P(y_1,\ldots,y_n|\mathbf{x}_1,\ldots,\mathbf{x}_n)} \quad (A.6)$$

Using (A.1), (A.3), and (A.6), we derive the following posterior probabilities of the weights:

$$P(\mathbf{w}|(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_n,y_n) \propto \exp\left(-(\frac{1}{\sigma_w^2}\sum_{j=1}^{N} w_j^2 + \frac{1}{\sigma_o^2}\sum_{i=1}^{n}[y_i - f(\mathbf{x}_i,\mathbf{w})]^2)\right) \quad (A.7)$$

$$\propto \exp\left(-(\sum_{i=1}^{n}[y_i - f(\mathbf{x}_i,\mathbf{w})]^2 + \lambda\sum_{j=1}^{N} w_j^2)\right) \quad (A.8)$$

where λ equals σ_o^2/σ_w^2.

In $E(\mathbf{w}) = \sum_{i=1}^{n}[y_i - f(\mathbf{x}_i,\mathbf{w})]^2 + \lambda\sum_j w_j^2$ we recognise the well-known cost function for neural network learning with weight decay term λ, which was introduced in Chapter 3. So the most probable or maximum posterior weights are identical to the weights obtained by minimising the cost function E. The inclusion of the penalty term (weight decay term) λ reduces the tendency of maximum likelihood estimation to "overfit" the data, i.e., to model the noise rather than the true regularities.

A difficulty arises when specifying the value of λ. In practice it is often impossible to specify λ $(= \sigma_o^2/\sigma_w^2)$ *a priori*. Different authors choose different approaches to this specification problem. One approach uses some approximation of the prediction performance to select a suitable value for the weight decay term, for instance by cross-validation. Other approaches, which are in line with the Bayesian theory, specify a non-informative prior for the parameters σ_o and σ_w. An approach followed by MacKay [Mac92] is to estimate the parameters from the data during learning.

A.2. The Predictive Approach

In the *predictive* approach to statistical prediction, one does not use a single "best" weight vector, but integrates over the posterior weight distribution. The best single-valued prediction for a test case with input $\mathbf{x}_{n+1}$ is then given by

$$\hat{y}_{n+1} = \int_{\mathcal{R}^N} f(\mathbf{x}_{n+1},\mathbf{w})P(\mathbf{w}|(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_n,y_n))d\mathbf{w} \quad (A.9)$$

where N is the dimension of the weight vector.

In large samples we expect the posterior distribution of the weights to concentrate near a single point, in this case the plug-in and the predictive approach become equivalent. For

small data samples there are trade-offs between the flexibility of $f(\mathbf{x}, \mathbf{w})$ and the spread of the posterior distribution of the weights $P(\mathbf{w}|(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n))$; see [GBD92]. A limited data set together with a flexible estimator leave room for numerous weight vectors to be probable. These weight vectors may be slightly less probable than the weights obtained by maximum likelihood estimation. To improve the average prediction performance, it looks promising to take the whole posterior distribution of the weights into account, instead of using only the single most probable weight vector for a particular data sample.

The integration of predictions is one way in which the Bayesian approach reduces overfitting. The other way is the preference over weights embodied in the prior. The integration of predictions makes a weight vector which fits the data only slightly better than other weight vectors contribute only slightly more to the actual prediction than the others. This contrasts with maximum likelihood estimation, in which the best weight vector dominates all the others. In this way the uncertainty in the determination of the best weight vector is explicitly taken into account. In practice, however, application of the foregoing is infeasible due to the high-dimensional integrals, which are analytically intractable and difficult to compute numerically.

At this point approximations are made. Researchers dedicated to Bayesian neural network learning differ in the way they handle high-dimensional integrals; see [Mac92], [Nea92], [BW91], [Rip93a], and [Tho93a]. In practical applications of neural network learning techniques, it is important for an approximation to be easy to implement.

Ripley proposes in [Rip93a] the following approximation. Let $E(\mathbf{w})$ denote the sum of the log-likelihood and the log-prior (the regularisation term), and H the Hessian of $E(\mathbf{w})$ at a local minimum $\mathbf{w}^*$. Locally the posterior distribution of the weights is approximated by Taylor expansion:

$$\exp -E(\mathbf{w}) \approx \exp - \left[E(\mathbf{w}^*) + (\mathbf{w} - \mathbf{w}^*)^T H(\mathbf{w} - \mathbf{w}^*)/2\right]. \tag{A.10}$$

Find as many local minima $\mathbf{w}_i^*$ $(i = 1, \ldots, q)$ of the cost function $E(\mathbf{w})$ as possible, and use the approximation (A.10). This combined with the lemma

$$\int \exp(-\frac{1}{2}\mathbf{w}^T H \mathbf{w}) d^N \mathbf{w} = (2\pi)^{N/2} \frac{1}{\sqrt{\det H}}$$

leads to the following approximation to (A.9):

$$\hat{y}_{n+1} = \sum_{i=1}^{q} f(\mathbf{x}_{n+1}, \mathbf{w}_i^*) I(\mathbf{w}_i^*), \tag{A.11}$$

where $I(\mathbf{w}_i^*)$ is proportionally to

$$\frac{1}{\sqrt{\det H}} \exp(-E(\mathbf{w}_i^*)).$$

This approach, although computationally very demanding, seems feasible in small sample situations when very good predictions are required. When, however, for a research activity many different models have to be fitted and compared, the computing time required for the predictive approach is yet prohibitively large.

Appendix B

Appendix to Chapter 8

B.1. Data Sources

Annual Data

The following yearly data series[1] have been used:

- M^{in}, The amount of *new* mortgages taken out on dwellings (Mln Dfl.); period 1965-1994; source CBS.
- M, The total stock of mortgages (Mln Dfl.); period 1965-1994; source CBS.
- DY, Disposable national income, gross and at market prices (Mln Dfl.); period 1965-1993; source CBS.
- NH, The total number of households (*1000); period 1965-1993; source CBS.
- ND, The total number of dwellings, rented and privately owned (*1000); period 1965-1992; source CBS.
- $\%NOH$, Percentage of total housing stock that is owner occupied (percentage); period 1965-1990 (5 yearly); source CPB.
- r_m, The average interest rate of new mortgages on real estate (percentage); period 1965-1994; source CBS.
- P, The price index of total consumption of employee households with an income less than the sick-fund limit (index); period 1965-1993; source CBS.
- P^H, The mean market price of empty to accept dwellings (*1000 Dfl.); period 1975-1994; source NVM.
- P^{NH}, The price index of new dwellings, incl. TAV (index); period 1965-1990; source CBS.

[1] The author wants to thank the ABN/AMRO bank for kindly providing these data.

- P^r, The price index of the rent of employee households (index); period 1965-1993; source CBS.
- WED, The number of weddings (*1000); period 1965-1994; source CBS.

These data originate from the CBS (Central Bureau of Statistics), the CPB (Central Planning Bureau), and the NVM (Dutch society of real estate agents). Some of the series have missing values for 1993 or 1994. We predicted these missing values, using a simple univariate time series model. On the important variable P^H, we have data starting at 1975; the missing data for the period 1965-1974 were approximated by deflating the 1975 market price for housing by the price index of new dwellings P^{NH} in the corresponding years, assuming that in those years the market price for housing followed the price changes of new dwellings. Finally, the stock of owner occupied houses (*1000) is calculated by multiplying the total number of dwellings ND by the percentage of the housing stock which is privately owned $\%NOH$. The latter variable, which is measured each five years, is transferred into yearly observations by a cubic spline interpolation.

Monthly Data

The following monthly data series have been used.

- M^{in}, The amount of *new* mortgages taken out on dwellings (Mln Dfl.); period 1985.01-1994.12; source ABN/AMRO.
- r_m, mortgage loan rate for 5 years fixed (percentage); period 1985.01-1994.12; source ABN/AMRO.
- H^s, total number of (existing) houses sold by the NVM (number); period 1985.01-1994.12; source NVM.
- P^H, market price of houses (Dfl.); period 1985.01-1992.12; source NVM; period 1992.01-1994.12; source land's register.
- $M(-1)$, total stock of mortgages in the previous year (Mln Dfl.); period 1985.01-1994.12; source CBS.

The market price for houses registered by the NVM is above the average market price. We have approximated the NVM data on the market price of houses for the 24 missing months by the data from the land's register, since this series ends at December 1992. In fact, we have inflated the last available NVM house price by a price index derived from the data on house prices taken from the land's register (with December 1992 as base).

Appendix C

Appendix to Chapter 9

C.1. Data Sources

Table C.1 presents the data source of the variables which are used in the structural exchange rate models. The first column gives the variable's symbol, the second column the variable's description, the third column the variable's measurement unit, the fourth column indicates the published data series it originates from, and the last column refers to the DATASTREAM code. To obtain data series of considerable length, we had to switch the money supply definition M1 to M0 for the United Kingdom case.

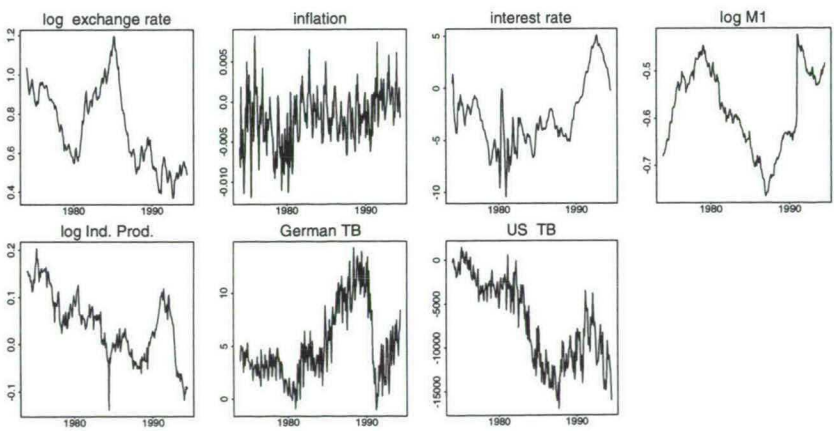

Figure C.1: Data for Germany-US; all data (except for exchange rate and trade balances) are in differential form

Figures C.1 through C.4 display the time paths of the variables that occur in the structural exchange rate models. These monthly series start in January 1974 and end in June 1994.

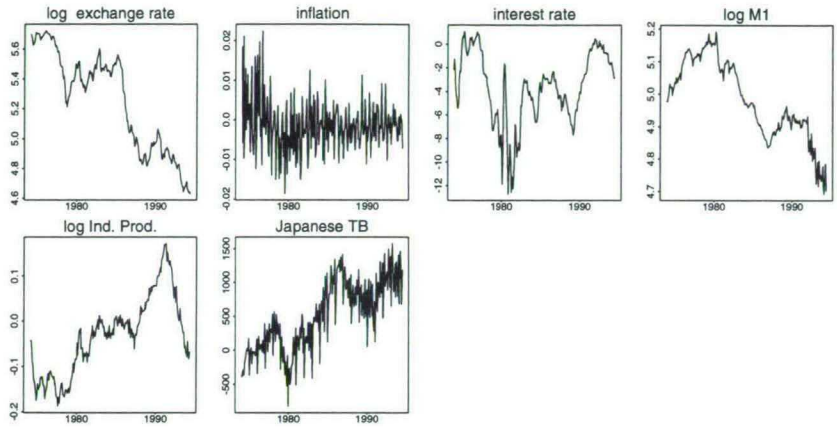

Figure C.2: Data for Japan-US; all data (except for exchange rate and trade balances) are in differential form

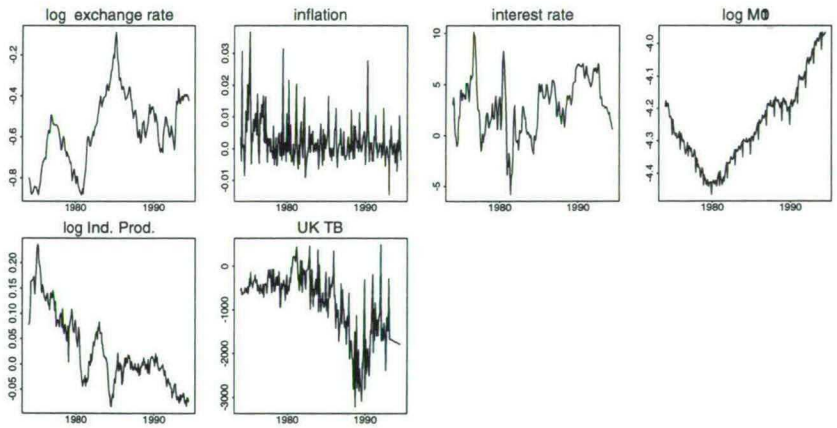

Figure C.3: Data for United Kingdom-US; all data (except for exchange rate and trade balances) are in differential form

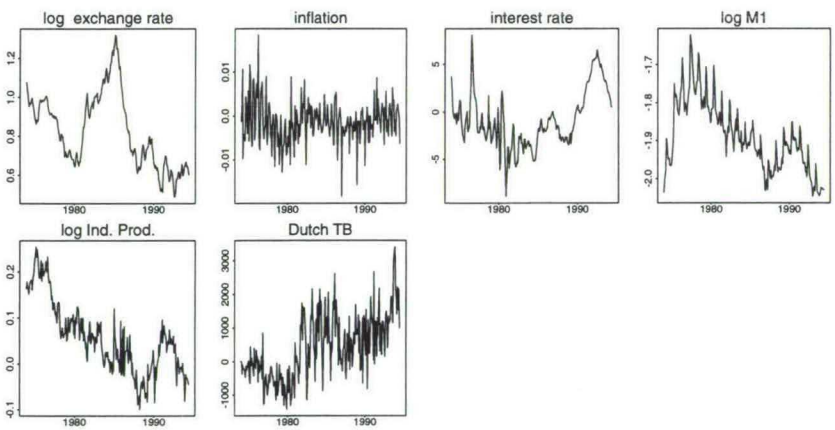

Figure C.4: Data for Netherlands-US; all data (except for exchange rate and trade balances) are in differential form

Table C.1: Data sources

variable	description	unit	series	code
	United States			
cpi	Consumer Prices	Index	OECD	USOCPCONF
r_s	short-term interest rate	Percentage	OECD	USOCTBL%
r_l	long-term interest rate	Percentage	OECD	USOCLNG%
m	money supply M1	US $ Bln	OECD	USOCM1MNA
	money supply M1	US $ Bln	OECD	USOCM1MNB
	monetary base M0	US $ Bln	GOV	USMONBASA
ip	industrial production -total	Index	OECD	USOCIPRDG
TB	Foreign Trade Balance	US $ Mln	OECD	USOCVBALA
	Germany			
cpi	Consumer Prices	Index	OECD	BDOCPCONF
r_s	short-term interest rate	Percentage	OECD	BDOCTBL%
r_l	long-term interest rate	Percentage	OECD	BDOCLNG%
m	money supply M1	DM Bln	OECD	BDOCM1MNB
ip	industrial production -total	Index	OECD	BDOCIPRDG
TB	Foreign Trade Balance	DM Bln	OECD	BDOCVBALA
	United Kingdom			
s	exchange rate -Pound to 1 US $		GOV	USX$UK..
cpi	Consumer Prices	Index	OECD	UKOCPCONF
r_s	short-term interest rate	Percentage	OECD	UKOCTBL%
r_l	long-term interest rate	Percentage	OECD	UKOCLNG%
m	money supply M0	Pound Bln	GOV	UKM0....A
ip	industrial production -total	Index	OECD	UKOCIPRDG
TB	Foreign Trade Balance	Pound Mln	OECD	UKOCVBALA
	Netherlands			
s	exchange rate -DFL to 1 US $		GOV	USX$DFL
cpi	Consumer Prices	Index	OECD	NLOCPCONF
r_s	short-term interest rate	Percentage	GOV	NLEURO3
r_l	long-term interest rate	Percentage	IMF	NLI61...
m	money supply M1	DFL Bln	OECD	NLOCM1MNA
ip	industrial production -total	Index	OECD	NLOCIPRDG
TB	Foreign Trade Balance	DFL Mln	OECD	NLOCVBALA
	Japan			
s	exchange rate -Yen to 1 US $		GOV	USX$YEN
cpi	Consumer Prices	Index	OECD	JPOCPCONF
r_s	short-term interest rate	Percentage	OECD	JPOCTBL%
r_l	long-term interest rate	Percentage	OECD	JPOCLNG%
m	money supply M1	Yen Bln	OECD	JPOCM1MNB
ip	industrial production -total	Index	OECD	JPOCIPRDG
TB	Foreign Trade Balance	Yen Mln	OECD	JPOCVBALA

note: M0 is the money base; M1 adds money of account to M0

C.2. Unit Root Test Results

Table C.2 presents the results of the ADF unit root tests for the variables in the exchange rate models for each country (The tests have been performed in PcGive 8.0). Recall from Chapter 5 that the null hypothesis of a unit root implies $\gamma_0 = 0$.

The first column of Table C.2 refers to the variable in differential form (except for the exchange rates and the trade balances); for example, r denotes $r - r^*$. The second through fourth columns give the "t-value" (negative sign omitted) of γ_0 in the following three transformed regressions

$$\Delta y_t = \alpha_0 + \alpha_1 t + \gamma_0 y_{t-1} + \sum_{i=1}^{p} \gamma_i \Delta y_{t-i} + \nu_t, \tag{C.1}$$

$$\Delta y_t = \alpha_0 + \gamma_0 y_{t-1} + \sum_{i=1}^{p} \gamma_i \Delta y_{t-i} + \nu_t, \tag{C.2}$$

$$\Delta y_t = \gamma_0 y_{t-1} + \sum_{i=1}^{p} \gamma_i \Delta y_{t-i} + \nu_t. \tag{C.3}$$

(C.4)

The number of lags p is determined by the highest possible lag (with a maximum of 13) which is significant at an 10 % error level. The corresponding critical values at the 1%, 5%, and 10% error levels are calculated following MacKinnon [Mac91], and are displayed at the bottom of each column. The last two columns give the t-values of α_0 and α_1 in either

$$\Delta y_t = \alpha_0 + \alpha_1 t + \sum_{i=1}^{p} \gamma_i \Delta y_{t-i} + \nu_t$$

if the null hypothesis of $\gamma_0 = 0$ could *not* be rejected, and in (C.2), (C.3), or (C.4), if the null could be rejected. Substituting $\gamma_0 = 0$ into (C.2) removes the possible multicollinearity between the trend and y_{t-1}, which makes the estimation of α_0 and α_1 more accurate.

Table C.2: Unit root tests

variable	$t_{\gamma_0}^{(1)}$	$t_{\gamma_0}^{(2)}$	$t_{\gamma_0}^{(3)}$	t_{α_0}	t_{α_1}
United Kingdom-US					
s	2.44	2.45	1.37	0.41	0.58
r	2.71	2.64	1.67	0.04	0.16
Δcpi	3.39	2.74	2.62**		
m	2.26	0.24	1.42	0.69	1.33
ip	3.95*	2.91*	3.27**	2.44*	2.61**
TB_{uk}	2.25	0.83	0.06	0.13	0.50
TB_{us}	1.95	0.85	0.66	0.52	0.26
Netherlands-US					
s	1.78	1.25	0.79	0.13	0.26
r	2.07	1.54	1.25	0.56	0.50
Δcpi	2.26	2.27	1.70	0.74	0.75
m	4.40**	1.60	0.34	4.17**	3.96**
ip	2.50	1.90	2.39**		
TB	1.83	0.58	0.03	0.80	0.33
Germany-US					
s	1.80	1.25	0.93	1.07	0.69
r	2.06	1.23	0.86	0.93	1.20
Δcpi	1.96	1.54	1.46	0.13	0.14
m	1.37	1.44	0.65	0.22	0.47
ip	2.88	2.04	2.28*		
TB	2.09	1.77	0.75	0.07	0.17
Japan-US					
s	2.41	0.50	1.62	0.40	0.45
r	2.16	2.08	1.05	0.73	0.73
Δcpi	2.34	2.44	1.58	1.26	1.10
m	2.73	0.35	1.11	0.50	1.13
ip	0.90	1.60	1.60	1.27	1.29
TB	2.95	1.46	0.43	0.31	0.03
critical values:					
1%	4.00	3.46	2.57	2.60	2.60
5%	3.43	2.87	1.94	1.97	1.97

C.3. Econometric Analysis of Short-Run Models

The following procedure has been followed to arrive at a parsimonious short-run model for Δs_t. According to the general-to-specific approach, discussed in Chapter 1, we depart from the most

general model (portfolio) extended with three lags for each variable included. We then use F-tests to test for zero restrictions on a subset of variables. First the cumulated trade balance terms are tested on their relevance, next the inflation rate differentials, and finally the elements of the flexible price model (r, m, and ip). Only the end-results of this 'testing down' process are reported in Table C.3. Table C.3 reports the variable's name, its coefficient, its t-value, the

Table C.3: Model Estimates

variable	coefficient	t-value
Japan (R^2=0.19; DW=1.97)		
Δs_{t-1}	0.36	6.11
Δs_{t-3}	0.10	1.76
Δm_{t-2}	0.17	1.94
Δr_t	-0.0075	-3.96
United Kingdom (R^2=0.25; DW=2.00)		
Δs_{t-1}	0.47	7.70
Δs_{t-2}	-0.16	-2.53
Δm_{t-2}	0.20	2.43
ΔTB_t^*	3.07e-6	3.54
ΔTB_t	-7.91e-6	-2.58
ΔTB_{t-2}	-1.18e-5	-3.31
Germany (R^2=0.21; DW=1.96)		
Δs_{t-1}	0.34	5.99
Δm_{t-2}	0.29	2.76
Δr_t	-0.0086	-4.55
Δip_{t-1}	-0.19	-2.39
The Netherlands (R^2=0.20; DW=1.99)		
Δs_{t-1}	0.34	5.90
Δm_{t-1}	0.13	2.51
Δm_{t-3}	0.11	2.20
Δr_t	-0.0064	-4.27

R^2 of the estimated model, and the Durbin-Watson statistic DW, defined as

$$DW := \frac{\sum_{t=2}^{n}(e_t - e_{t-1})^2}{\sum_{t=1}^{n} e_t^2},$$

where e denotes the observed residual of the estimated model. A value of the DW-statistic close to 2 indicates no autocorrelation in the residuals. Some other diagnostic tests have been performed as well.

Note that the dominant factor in all models is Δs_{t-1}. Recursive estimation showed that most parameter estimates were stable, with some exceptions for the money supply variables. Stable parameter estimates are a prerequisite for reliable predictions.

We performed the neural network test (see section 6.2) to test for possible neglected nonlinearities in the models presented in Table C.3. The adjusted p-value of the test was smaller than 0.000 for the Japan, German, and UK cases; for the Dutch case the adjusted p-value was 0.92. Additionally, RESET tests, which add $\widehat{\Delta s_t}^2$ to the models, were performed. The probabilities of the observed F-statistics were 0.036, 0.136, 0.121, and 0.842 for Japan, Germany, U.K, and the Netherlands respectively. Both tests suggest that possible nonlinearities are present in the exchange rate models for Japan, Germany, and the U.K., and none in the model for the Netherlands. The evidence of the neural network test is stronger than that of the RESET test.

Samenvatting

Economische modellen geven een relatie weer tussen economische grootheden. Deze relatie moet de data uit de steekproef zo goed mogelijk benaderen en moet toekomstige waarnemingen zo accuraat mogelijk voorspellen. De steeds verder gaande ontwikkeling van de computer, met name op het gebied van rekensnelheid en geheugencapaciteit, maakt onderzoek naar en toepassing van rekenintensieve technieken voor datamodellering mogelijk. Deze technieken, ook wel flexibele regressietechnieken genoemd, zoeken in zeer grote klassen van functies naar een functioneel verband dat de observaties zo goed mogelijk benadert. Daarbij wordt *niet* een specifieke functionele vorm, bijvoorbeeld lineair, vooröndersteld. Een voorbeeld van een flexibele regressietechniek, die erg populair is op dit moment, is het neurale netwerk.

Deze studie is verdeeld in twee gedeelten: Theorie en Toepassingen. Part I (Theorie) dat de hoofdstukken 1-5 omvat, behandelt de theoretische aspekten van economisch modelleren en van neurale netwerken. Part II (Toepassingen) dat de hoodstukken 6-10 omvat, behandelt de praktische aspekten van het toepassen van neurale netwerken op problemen uit de economie.

Hoofdstuk 1 beschrijft het algemene proces van economisch modelleren in drie stappen: specificatie van het model, schatten van model parameters en evaluatie van het model. Aangenomen is dat alle modellen lineair of parametrisch zijn. De aan de orde gestelde onderwerpen betreffen zowel "cross-sectional" data als tijdreeksen. Het kernpunt van modelspecificatie is het bouwen van een kwantitatief economisch model, zonder daarbij de uitkomsten van klassieke statistische analyses betekenisloos te maken. De methodologie voor modelspecificatie is een veel besproken onderwerp in de econometrische literatuur. In dit proefschrift zijn de belangrijkste punten hieruit aangehaald. Daarnaast beschrijven we het schatten van de model parameters en het evalueren van een geschat model.

In Hoofdstuk 2 bespreken we een aantal flexibele regressiemethodieken uit de statistiek. Het neurale netwerk wordt geïntroduceerd als een element van deze algemene klasse. Een belangrijk probleem waarmee men bij de gehele klasse van flexibele regressietechnieken te maken heeft, is het zogenaamde zuiverheid/precisie dilemma. Een praktisch gevolg van dit dilemma is dat een nauwkeurige benadering van de steekproefdata vaak samengaat met een onnauwkeurige voorspelling van nieuwe observaties. Flexibele regressietechnieken hebben meestal een of meerdere (flexibiliteits)parameters, welke de mate van flexibiliteit (gladheid) van het resulterende be-

naderende verband beïnvloeden. Uit het zuiverheid/precisie dilemma volgt dat voorspellingen mogelijkerwijs verbeterd kunnen worden, wanneer de flexibiliteit van het benaderende verband enigszins ingeperkt wordt. Het instellen van de flexibiliteitsparameters gebeurt meestal op grond van een maat voor het generalisatievermogen, bijvoorbeeld de voorspelfout. In kleine steekproeven is het vaak moeilijk een betrouwbare schatting van de voorspelfout te maken. De statistische kruisvalidatie methode levert een schatting voor de voorspelfout in deze situaties.

Hoofdstuk 3 bespreekt neurale netwerken vanuit een statistisch theoretisch oogpunt. Het hoofdstuk begint met de grafische en wiskundige representatie van het neurale netwerk. Leren door neurale netwerken wordt besproken vanuit een statistisch perspektief. We laten zien dat wanneer de architectuur van het neurale netwerk bepaald is, dit leren conceptueel erg veel lijkt op niet-lineaire regressie. De grootste zorg die men heeft wanneer neurale netwerken worden toegepast op praktische problemen, is het bewerkstelligen van een acceptabel generalisatieniveau. Dit volgt direkt uit het zuiverheid/precisie dilemma. Om de generalisatiekwaliteit van een neuraal netwerk te verbeteren, is een regularisatiemethode geïntroduceerd, namelijk gewichtsverval ('weight decay'). Om te voorkomen dat de gewichten in het neurale netwerk te groot worden, wordt in deze methode bij de klassieke kwadratische verliesfunctie de gekwadrateerde som van de gewichten opgeteld. Een gangbare manier om de kwaliteit van een neuraal netwerk te beoordelen, is zijn voorspelfout te vergelijken met die van alternatieve technieken. Wanneer men echter statistisch verantwoorde conclusies wil trekken uit zulke vergelijkingen, dient men expliciet rekening te houden met het zogenaamde meervoudigheidseffect ('multiplicity effect'). Wij introduceerden enkele statistische methoden die ervoor zorgen dat verantwoorde conclusies getrokken kunnen worden uit een studie waarin meer dan twee regressiemethoden met elkaar worden vergeleken ('multiple comparisons').

Hoofdstuk 4 behandelt de belangrijkste praktische aspekten van neurale netwerken, zoals de specificatie van de architectuur van het netwerk, de verschillende onderdelen, de leerprocedure en de gebruikte software. Uit simulatie-experimenten blijkt dat meerdere lokaal optimale gewichtsvectoren vaak voorkomen, dat de methode van gewichtsverval de flexibiliteit van het resulterend verband effektief inperkt, evenals het aantal lokale minima. Om volledige openheid en duidelijk te betrachten in het construeren van een neuraal netwerk, hebben wij procedure geformuleerd voor de constructie van een neuraal netwerk. In deze procedure wordt op een consistente wijze het aantal verborgen neuronen bepaald, alsmede de waarde van de gewichtsverval-parameter.

In Hoofdstuk 5 wordt besproken waar en hoe neurale netwerken gebruikt kunnen worden bij het econometrische modelleren van niet-stationaire tijdreeksen. Gestart wordt met een intuïtieve verklaring van de gevolgen die niet-stationaire tijdreeksen kunnen hebben voor voorspellen. Daarna worden cointegratie en foutencorrectie modellen geïntroduceerd. Deze econometrische concepten zijn ontwikkeld om op een verantwoorde manier modellen te ontwikkelen, wanneer

de variabelen gerepresenteerd worden door niet-stationaire tijdreeksen. Neurale netwerken zijn gebruikt om een niet-lineaire generalisatie van cointegratie en fouten-correctie te operationaliseren. Kritieke waarden zijn gegenereerd door de 'augmented Dickey-Fuller' test voor neurale netwerken, die toetst op de aanwezigheid van niet-lineaire cointegratie. Het lineaire model met fouten-correctie is uitgebreid op een niet-lineaire manier, door in het gedeelte voor de korte termijn een geparametriseerd neuraal netwerk op te nemen.

Hoofdstuk 6 vormt de overgang van de *techniek* van neurale netwerken naar het *toepassen* van neurale netwerken binnen het economische en financiële domein. Het doel van dit hoofdstuk is om aan de hand van een overzicht van de literatuur de problemen te karakteriseren waarop neurale netwerken toegepast kunnen worden. Neurale netwerken worden het meest toegepast op zogenaamde klassificatieproblemen, zoals kredietbeoordeling en het waarderen van aandelen van een bepaald bedrijf. In veel van de onderzochte studies is het onduidelijk welke methodologie precies gevolgd is om tot de uiteindelijk architectuur van een neuraal netwerk te komen; bovendien worden soms conclusies getrokken op basis van (te) kleine steekproeven.

In Hoofdstuk 7 zijn neurale netwerken toegepast om een (zgn. hedonistisch) model voor de huisprijs voor woningen in Boston (USA) te maken, aan de hand van een representatieve steekproef met 506 waarnemingen. Er zijn 13 mogelijk verklarende variabelen. Twee parametrische modellen en een neuraal netwerk model zijn geschat met data uit een willekeurig gekozen deelverzameling ter grootte 400. De voorspelfouten, welke gemeten zijn op de overige 106 waarnemingen, zijn statistisch met elkaar vergeleken door middel van paarsgewijze t-toetsen. Deze t-toetsen zijn gebaseerd op 'resampling' (herhaalde trekkingen) en zijn gecorrigeerd voor het meervoudigheidseffekt. Uit deze toetsen mogen we concluderen dat het neurale netwerk een significant lagere gemiddelde voorspelfout geeft dan de twee parametrische modellen. Omdat het neurale netwerk een betere representatie van het onderliggende systeem blijkt te geven, loont het de moeite om het gevonden verband nader te onderzoeken. Daartoe worden drie maten voorgesteld, die voor iedere verklarende variabele de gemiddelde invloed, de gemiddelde absolute invloed en de mate van monotoniciteit van de partiële relatie aangeven.

In Hoofdstuk 8 zijn neurale netwerken toegepast om een voorspelmodel voor de produktie van nieuwe hypotheekleningen in Nederland te maken. Het voorspelmodel moet zowel voorspellingen kunnen genereren voor 1 maand vooruit als voor 18 maanden vooruit. De economische grootheden die een rol spelen in dit probleem, worden gekenmerkt als niet-stationaire tijdreeksen en worden met verschillende tijdsfrequenties gemeten (maandelijks of jaarlijks). Het probleem is aangepakt met een model met fouten-correctie waarin aspekten van de lange en korte termijn met elkaar gecombineerd worden. De lange-termijn-component is benaderd door een linear model gebaseerd op jaardata. Het neurale netwerk is gebruikt om te onderzoeken of een niet-lineaire specificatie van de korte-termijn-component de voorspelkwaliteit van het ECM verbetert. In deze studie was de verbetering echter verwaarloosbaar.

In Hoofdstuk 9 zijn neurale netwerken toegepast om te onderzoeken of het toevoegen van niet-lineariteiten aan bestaande strukturele wisselkoersmodellen de voorspelkwaliteit verbetert. We hebben drie wisselkoersmodellen onderzocht voor de volgende vier wisselkoersen: nederlandse gulden/amerikaanse dollar, japanse yen/amerikaanse dollar, britse pond/amerikaanse dollar en duitse mark/amerikaanse dollar. De drie theoretische wisselkoersmodellen zijn: het monetaire model met flexibele prijzen, het monetaire model met vaste prijzen en het model gebaseerd op een evenwichtige portefeuille. Uit deze theoretische modellen zijn empirische modellen geformuleerd. De empirische modellen zijn vervolgens gespecificeerd door een lineair verband en door een verband gebaseerd op een neuraal netwerk. De voorspelkwaliteiten van de verschillende modellen (in niveaus) zijn vergeleken met het 'random walk'-model voor de lange en korte termijn. De belangrijkste conclusie is dat voor de lange termijn de voorspelkwaliteit (tot een voorspelhorizon van 2 jaar) slechter is dan die van het 'random walk- model; dit geldt ook voor de neurale netwerken. Wanneer de verandering in de wisselkoers voorspeld wordt, blijken de veranderingen niet geheel random te zijn. Een eenvoudig lineair model waarin de verandering van de wisselkoers in de vorige maand een belangrijke rol speelt, voorspelt iets beter dan het 'random walk'-model (in niveaus). Een neuraal netwerk vindt geen specificatie die betere voorspellingen oplevert.

In Appendix A wordt het leren van een neuraal netwerk en het maken van voorspellingen met een neuraal netwerk besproken vanuit een Bayesiaans perspektief. De Bayesiaanse interpretatie van de gewichtsverval-parameter uit Hoofdstuk 3 is een prior-verdeling op de gewichten.

Appendix B geeft de beschrijving en herkomst van de variabelen die zijn gebruikt in Hoofdstuk 8.

Appendix C geeft de beschrijving en herkomst van de variabelen die zijn gebruikt in Hoofdstuk 9. Verder zijn hier de numerieke resultaten van enkele, in Hoofdstuk 9, gebruikte toetsen en analyses weergegeven.

Uit het gedane onderzoek concluderen we het volgende. Neurale netwerken kunnen handig toegepast worden op verschillende economische modelleringsproblemen. Ze kunnen worden ingebed in de binnen de economie algemeen geaccepteerde empirische onderzoeksmethodologie. Wij hebben een strategie voor het maken van een neuraal netwerk ontwikkeld die vanzelf aangeeft of niet-lineaire benaderingen van de waarnemingen gerechtvaardigd zijn. Op deze manier wordt misspecificatie van de functionele vorm nagenoeg uitgeschakeld als oorzaak voor (eventuele) slechte prestaties van het model. *Neurale netwerken vormen derhalve een nuttige toevoeging aan het arsenaal van econometrische methoden, maar zijn geen vervanging van bestaande modellerings- en analysemethoden.*

Bibliography

[AH84] G. Anderson and D. Hendry. An econometric model of United Kingdom building societies. *Oxford Bulletin of Economics and Statistics*, 46(3):185–210, 1984.

[AK89] E. Aarts and J. Korst. *Simulated annealing and Boltzmann machines : a stochastic approach to combinatorial optimization and neural computing*. Wiley, Chichester, 1989.

[Alt92] N.S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46:175–185, 1992.

[ARe88] J. Anderson and E. Rosenfeld (eds.). *Neurocomputing: Foundations of Research*. MIT Press, Cambridge, 1988.

[Arm78] S. Armstrong. *Long-range forecasting: from crystal ball to computer*. Wiley & Sons, New York, 1978.

[BDGH93] A. Banerjee, J. Dolado, J. Galbraith, and D. Hendry. *Co-integration, Error-correction, and the Econometric Analysis of Non-stationary Data*. Oxford University Press, Oxford, 1993.

[Bel61] R. Bellman. *Adaptive Control Processes: A guided tour*. Princeton University Press, Princeton, New Jersey, 1961.

[Ber85] J.O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, New York, 1985.

[Ber90] R. Berk. A primer on rubust regression. In J. Fox and J. Long, editors, *Modern methods of data analysis*, pages 292–324. Sage, Newbury Park, CA, 1990.

[BFOS84] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks/Cole, Monterey, CA, 1984.

[BH85] W. Branson and D. Henderson. The specification and influence of asset markets. In R.W. Jones and P.B. Kenen, editors, *Handbook of International Economics, vol. 2*, pages 749–805. North Holland, Amsterdam, 1985.

[Bil78] J.F. Bilson. Rational expectations and the exchange rate. In J. Frenkel and H. Johnson, editors, *The economics of exchange rates*. Addison-Wesley Press, Reading, 1978.

[BKW80] D. Belsley, E. Kuh, and R. Welsch. *Regression diagnostics: Identifying influential data and sources of collinearity*. John Wiley & Sons, New York, 1980.

[BKW93] A. Bansal, R. Kaufmann, and R. Weitz. Comparing the modelling performance of regression and neural networks as data quality varies: A business value approach. *Journal of Management Information Systems*, 10(1):11–32, 1993.

[BM78] C. Beach and G. MacKinnon. A maximum likelihood procedure for regression with autocorrelated errors. *Econometrica*, 46(1):17–34, 1978.

[BM89] R. Baillie and P. McMahon. *The foreign exchange market: theory and econometric evidence*. Cambridge University Press, Cambridge, 1989.

[BRe94] B. Bhaskara Rao (ed.). *Cointegration for the applied economist*. St. Martin's Press, New York, 1994.

[Bur92] Central Planning Bureau. *A macro-econometric model for the Netherlands*. Stenfert Kroese, Leiden, 1992.

[BvdBW94] D. Baestaens, W. van den Bergh, and D. Wood. *Neural Network Solutions for Trading in Financial Markets*. Pitman, London, 1994.

[BW91] W.L. Buntine and A.S. Weigend. Bayesian back-propagation. *Complex Systems*, 5:603–643, 1991.

[Cag56] P. Cagan. In M. Friedman, editor, *Studies in the Quantitative Theory of Money*. University of Chicago Press, Chicago, 1956.

[CD92] W.W. Charemza and D.F. Deadman. *New directions in econometric practice*. Edward Elgar, Brookfield, 1992.

[CH90] M.J. Crowder and D.J. Hand. *Analysis of repeated measures*. Chapman & Hall, London, 1990.

[CO94] P. Cheeseman and R.W. Oldford, editors. *Selecting models from data: AI and statistics IV.* Lecture notes in statistics nr. 89. Springer-Verlag, New York, 1994.

[Col91] A. Collard. A B-P ANN commodity trader. In R. Lippmann, J. Moody, and D. Touretzky, editors, *Advances in Neural Information Processing Systems 3*, pages 551–556. 1991.

[CT95] M. Camarero and C. Tamarit. A rationale for macroeconomic policy coordination: evidence based on the spanish peseta. *European Journal of Political Economy*, 11:65–82, 1995.

[Cyb89] Cybenko. Approximation by superposition of a sigmoidal function. *Math. Control Systems Signals*, 2:303–314, 1989.

[DD88] Bates D.M. and Watts D.G. *Nonlinear regression analysis and its applications.* John Wiley & Sons, New York, 1988.

[DDG94] C.G. Dasgupta, G. Dispensa, and S. Ghose. Comparing the predictive performance of neural network models with some traditional market response models. *International journal of Forecasting*, 10:235–244, 1994.

[DKK93] R. Donaldson, M. Kamstra, and Kim. Evaluating alternative models for the conditional volatility of stock returns: evidence from international data. Technical report, University of British Columbia, 1993.

[DKV95] H. Daniels, B. Kamp, and W. Verkooijen. Design of neural networks for prediction and classification in economic problems. In *Proceedings on IFAC Symposium on modelling and control of national and regional economies.* Gold Coast, Australia, 1995.

[DN90] F. Diebold and J. Nason. Nonparametric exchange rate prediction? *Journal of International Economics*, 28:315–332, 1990.

[Dor76] R. Dornbusch. Expectations and exchange rate dynamics. *Journal of Political Economy*, 84:1161–1176, 1976.

[DS94] S. Dutta and S. Shekhar. Bond rating: a non-conservative application of neural networks. In R. Trippi and E. Turban, editors, *Neural networks in finance and investing*, pages 257–273. Probus Publishing, 1994.

[Efr83] B. Efron. Estimating the error rate of a prediction rule. *J. Amer. Statist. Assoc.*, 78:316–333, 1983.

[EGH91] R. Engle, C. Granger, and J. Hallman. Merging short- and long-run forecasts: An application of seasonal cointegration to monthly electricity sales forecasting. In R. Engle and C. Granger, editors, *Long-Run Economic Relationships*, pages 220–236. Oxford University Press, 1991.

[EY87] R. Engle and S. Yoo. Forecasting and testing in co-integrated systems. *Journal of Econometrics*, 35:143–159, 1987.

[Fah88] S. Fahlman. An empirical study of learning speed in back-propagation networks. Technical Report CMU-CS-88-162, 1988.

[FF93] I.E. Frank and J.H. Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, 35(2):109–135, 1993.

[FG93] D. Fletcher and E. Goss. Forecasting with neural networks: an application using bankruptcy data. *Information & Management*, 24:159–167, 1993.

[FHZ93] W. Finnoff, F. Hergert, and H. Zimmermann. Improving model selection by nonconvergent methods. *Neural Networks*, 6:771–783, 1993.

[Fis35] R.A. Fisher. *The Design of Experiments*. Olivier & Boyd, Edinburgh, 1935.

[FKB90] M. Fase, P. Kramer, and W. Boeschoten. *MORKMON II: het DNB kwartaalmodel voor Nederland*. De Nederlandsche Bank NV, Amsterdam, 1990.

[FL90] S. Fahlman and C. Lebiere. The cascade-correlation learning algorithm. In D. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 525–532. 1990.

[Fle62] J.M. Fleming. Domestic financial policies under fixed and floating exchange rates. *International Monetary Fund Staff Papers*, 9(4):369–380, 1962.

[Fra79] J.A. Frankel. On the mark: A theory of floating exchange rates based on real interest differentials. *The American Economic Review*, 69:611–622, 1979.

[Fre76] J.A. Frenkel. A monetary approach to the exchange rate: doctrinal aspects and empirical evidence. *Scandinavian journal of economics*, 76:200–224, 1976.

[Fre81] J.A. Frenkel. The collapse of purchasing power parities during the 1970's. *European Economic Review*, 16(4):145–165, 1981.

[Fre94] J. Freeman. *Simulating Neural Networks with Mathematica*. Addison Wesley, New York, 1994.

[Fri91] J.H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19:1–141, 1991.

[FS81] J. Friedman and W. Stuetzle. Projection pursuit regression. *J. Americ. Statist. Soc.*, 76:817–823, 1981.

[Ful76] W.A. Fuller. *Introduction to Statistical time series*. Wiley, New York, 1976.

[FV95] A. Feelders and W. Verkooijen. On the statistical comparison of inductive learning methods. In *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics, 4-7 Jan., Ft. Lauderdale, Florida*. Springer Verlag, 1995.

[GBD92] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58, 1992.

[GFR94] W. Goffe, G. Ferrier, and J. Rogers. Global optimization of statistical functions with simulated annealing. *Journal of econometrics*, 60(1):65–100, 1994.

[GH91a] C. Granger and J. Hallman. Long memory series with attractors. *Oxford Bulletin of Economics and Statistics*, 53(1):11–26, 1991.

[GH91b] C. Granger and J. Hallman. Nonlinear transformations of integrated time series. *Journal of Time Series Analysis*, 12(3):207–224, 1991.

[GNS94] W. Gorr, D. Nagin, and J. Szczypula. Comparative study of artificial neural network and statistical models for predicting student grade points averages. *International Journal of Forecasting*, 10:17–34, 1994.

[GO93] G. Grudnitski and L. Osburn. Forecasting S&P and gold futures prices: An application of neural networks. *Journal of Futures Markets*, 13(6):631–643, 1993.

[Gol89] D. E. Goldberg. *Genetic Algorithms in search, optimization, and machine learning*. Addison-Wesley, Reading, MA, 1989.

[Gra90] C. Granger. *Modelling Economic Series: Readings in Econometric Methodology*. Oxford University Press, Oxford, 1990.

[Gra94] Clive W.J. Granger. Forecasting in economics. In A.S. Weigend and N.A. Gershenfield, editors, *Time Series Prediction: Forecasting the Future and Understanding the Past*, pages 529–539. Addison-Wesley, Reading, MA, 1994.

[Gre93] W. Greene. *Econometric Analysis*. Macmillan, New York, 1993.

[GT93] C. Granger and T. Teräsvirta. *Modelling nonlinear economic relationships*. Oxford University Press, Oxford, 1993.

[Had76] Hadjimatheou. *Housing and mortgage markets: the UK experience*. Saxon House, Farnborough, 1976.

[Hae90] W. Haerdle. *Applied nonparametric regression*. Cambridge University Press, Cambridge, 1990.

[Hal94] S. Hall. *Applied Economic Forecasting Techniques*. Harvester Wheatsheaf, New York, 1994.

[Han93] D. J. Hand, editor. *Artificial intelligence frontiers in statistics: AI and statistics III*. Chapman & Hall, London, 1993.

[Har90] A. Harvey. *The Econometric Analysis of Time Series*. Philip Allan, New York, 1990.

[Hay88] W. Hays. *Statistics*. Holt, Rinehart and Winston, Inc, Fort Worth, 1988.

[Hen93] D. Hendry. *Econometrics: Alchemy or Science?* Blackwell, Oxford, 1993.

[HJKS92] A. Hut, Y. Jurriens, J. Kikstra, and F. Suijker. De woningmarkt op de lange termijn. Technical Report 98, Centraal Planbureau, juni 1992.

[HJR90] D. Hawley, J. Johnson, and D. Raina. Artificial neural systems: A new tool for financial decision making. *Financial Analysts Journal*, pages 63–72, 1990.

[HKP91] J. Hertz, A. Krogh, and R. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley, Redwood City, CA, 1991.

[HLMS93] J. Hwang, D. Li, M. Maechler, and J. Schimmert. Regression modelling in back-propagation and projection pursuit learning. *IEEE Trans. Neural Networks*, 1993.

[HMOR94] T. Hill, L. Marquez, M. O'Connor, and W. Remus. Artificial neural network models for forecasting and decision making. *international journal of forecasting*, 10:5–15, 1994.

[HN90] R. Hecht-Nielsen. *Neurocomputing*. Addison-Wesley, Reading, 1990.

[Hol92] M. Holmes. The demand for building society mortgage finance in northern Ireland and Scotland. Technical Report 92/3, Loughborough University of Technology, february 1992.

[Hop93] R. Hoptroff. The principles and practice of time series forecasting and business modelling using neural nets. *Neural Computing & applications*, 1:59–66, 1993.

[HOR94] T. Hill, M. O'Connor, and W. Remus. Neural networks for time series forecasting. Technical report, University of Hawaii, 1994.

[HP94] D. Holden and R. Perman. Unit roots and cointegration for the economist. In B. Bhaskara Rao, editor, *Cointegration for the applied economist*, pages 48–112. St. Martin's Press, 1994.

[HPT90] K. Holden, D. Peel, and J. Thompson. *Economic forecasting: an introduction.* Cambridge University Press, Cambridge, 1990.

[HR78] D. Harrison and D. Rubinfeld. Hedonic prices and the demand for clean air. *Journal of Environmental Economics and Management*, 53(5):81–102, 1978.

[HT87] Y. Hochberg and A. Tamhane. *Multiple comparison procedures.* Wiley & Sons, New York, 1987.

[HT90] T. Hastie and R. Tibshirani. *Generalized additive models.* Chapman and Hall, London, 1990.

[HU89] S. Hall and R. Urwin. A disequilibrium model of building society mortgage lending. Technical Report 26, Bank of England, July 1989.

[Hub85] P.J. Huber. Projection pursuit. *The Annals of Statistics*, 13:435–475, 1985.

[Jan92] J. Janssen. *De prijsvorming van bestaande koopwoningen.* PhD thesis, Katholieke Universiteit Nijmegen, 1992.

[JG85] G. Judge and W. Griffiths. *The theory and practice of econometrics.* Wiley & Sons, New York, 1985.

[Jon93] L.D. Jones. The demand for home mortgage debt. *Journal of Urban Economics*, 35:10–28, 1993.

[Keu94] H. Keuzenkamp. *Probability, Econometrics and Truth: A Treatise on the Foundations of Econometric Inference.* PhD thesis, Tilburg University, 1994.

[Kie93] J. Kiel. Een verslag: het maken van een model voor de hypotheekmarkt (*in dutch*). stageverslag, October 1993.

[KM94] H. Keuzenkamp and J. Magnus. On tests and significance in econometrics. Technical Report No. 9431, Center, March 1994.

[Koh95] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In Chris S. Mellish, editor, *Proceedings of IJCAI-95, Volume 2*, pages 1137–1143. Morgan Kaufmann, San Mateo (CA), 1995.

[KW92] C. Kuan and H. White. Artificial neural networks: An econometric perspective. *Econometric Reviews*, 13(1):1–91, 1992.

[KWR93] J. Kim, H. Weistroffer, and R. Redmond. Expert systems for bond rating: a comparative analysis of statistical, rule-based and neural network systems. *Expert Systems*, 10(3):167–171, 1993.

[lC89] Y. le Cun. Generalization and network design strategies. Technical Report CRG-TR-89-4, University of Toronto, June 1989.

[Lea78] E.E. Leamer. *Specification Searches: Ad Hoc Inference with Nonexperimental Data*. Wiley & Sons, New York, 1978.

[LWG93] T. Lee, H. White, and C. Granger. Testing for neglected nonlinearity in time series models: a comparison of neural network methods and alternative test. *Journal of Econometrics*, 56:269–290, 1993.

[Mac91] MacKinnon. Critical values for cointegration tests. In R. Engle and C. Granger, editors, *Long-Run Economic Relationships*, pages 267–276. Oxford University Press, 1991.

[Mac92] D.J. MacKay. Bayesian interpolation. *Neural Computation*, 4:415–447, 1992.

[Mar95] N.C. Mark. Exchange rates and fundamentals: Evidence on long-horizon predictability. *The American Economic Review*, 85(1):201–218, 1995.

[May79] D. Mayes. *The property boom: the effects building society behaviour on house prices*. Martin Robertson, Oxford, 1979.

[MCF+82] A. Makridakis, S. Anderson, R. Carbone, R. Fildes, R. Hibon, R. Lewandowski, J. Newton, E. Parzen, and R. Winkler. The accuracy of extrapolation (time series) methods: results of a forecasting competition. *Journal of forecasting*, 1:111–153, 1982.

[MF] D. Montgomery and D.J. Friedman. *IIE Transactions*, pages 73–85.

[MHWR94] L. Marquez, T. Hill, R. Worthley, and W. Remus. Neural network models as an alternative to regression. In R. Trippi and E. Turban, editors, *Neural networks in finance and investing*, pages 435–447. Probus Publishing, 1994.

[Mil90] A.J. Miller. *Subset selection in regression*. Chapman and Hall, London, 1990.

[MJ94] A. Murray and Ruggiero Jr. Training neural nets for intermarket analysis. *Futures*, pages 42–44, August 1994.

[MKA94] E. Maasoumi, A. Khotanzad, and A. Abaye. Artificial neural networks for some macroeconomic series: a first report. *econometric reviews*, 13(1):105–122, 1994.

[MP91] M. Meulepas and J. Plasmans. Estimating and forecasting exchange rates by means of ppp and uip. Technical Report SESO 91/262, UFSIA University of Antwerp, 1991.

[MR83] R. Meese and K. Rogoff. Empirical exchange rate models of the seventies: Do they fit out-of-sample? *Journal of International Economics*, 14:3–24, 1983.

[MR90] R. Meese and A. Rose. Non-linear, non-parametric, non-essential exchange rate determination. *American Economic Review*, 192-196:601–619, 1990.

[MR91] R. Meese and A. Rose. An empirical assessment of non-linearities in models of exchange rate determination. *Review of Economic Studies*, 58:601–619, 1991.

[MS91] R. Martin and D. Smyth. Adverse selection and moral hazard effects in the mortgage market: an empirical analysis. *Southern Economic Journal*, 57(4):1071–1084, 1991.

[MST94] D. Michie, D.J. Spiegelhalter, and C.C. Taylor, editors. *Machine learning, neural and statistical classification*. Ellis Horwood, New York, 1994.

[MT92] R. MacDonald and M. Taylor. Exchange rate economics: a survey. *IMF Staff Papers*, 39:1–57, 1992.

[MU92] J. Moody and J. Utans. Principled architecture selection for neural networks: application to corporate bond rating prediction. In J.E. Moody, S.J. Hanson, and R.P. Lippman, editors, *Advances in neural information processing systems 4*. Morgan Kaufmann Publishers, San Mateo, CA, 1992.

[MU94] J. Moody and J. Utans. Architecture selection strategies for neural networks: application to bond rating prediction. In A.N. Refenes, editor, *Neural Networks in the Capital Markets*. John Wiley & Sons, New York, 1994.

[Mun63] R.A. Mundell. Capital mobility and stabilization policy under fixed and flexible exchange rates. *Canadian Journal of Economic and Political Science*, 29(4):475–485, 1963.

[Mur94] M.P. Murray. A drunk and her dog: An illustration of cointegration and error correction. *The American Statistician*, 48:37–39, 1994.

[Nas90] J. Nash. *Compact Numerical Methods for Computers*. Adam Hilger, New York, 1990.

[Nea92] R. Neal. Bayesian training of backpropagation networks by the hybrid monte carlo method. Technical Report CRG-TR-92-1, University of Toronto, April 1992.

[Pag87] A. Pagan. Three econometric methodologies: A critical appraisal. *Journal of Econometric Survey*, 1:3–24, 1987.

[PDM+92] O. Pictet, M. Dacorogna, U. Müller, R. Olsen, and J. Ward. Real-time trading models for foreign exchange rates. *Neural Network World*, 2(6):713–744, 1992.

[Pet91] E. Peters. *Chaos and Order in the Capital Markets*. John Wiley& Sons, New York, 1991.

[Pra80] M. Pratt. Building societies: an econometric model. Technical Report 11, Bank of England, 1980.

[PT90] L.F. Pau and T. Tambo. Knowledge-based mortgage-loan credit granting and risk assessment. *Journal of Economic Dynamics and Control*, 14:255–262, 1990.

[RABCK93] A. Refenes, M. Azema-Barac, L. Chen, and S. Karoussos. Currency exchange rate prediction and neural network design strategies. *Neural Computing & Applications*, 1:46–58, 1993.

[Ree93] R. Reed. Pruning algorithms –a survey. *IEEE transactions on Neural Networks*, 4:740–747, 1993.

[RHW86] D. Rumelhart, G. Hinton, and R. Williams. Learning internal representations by error propagation. In D. Rumelhart and J. McClelland, editors, *Parallel distributed processing: explorations in the microstructures of cognition*, pages 318–362. MIT Press, Cambridge, MA, 1986.

[Rip93a] B.D. Ripley. Flexible non-linear approaches to classification. In V. Cherkassky, J.H. Friedman, and H. Wechsler, editors, to appear in: *From Statistics to Neural Networks: Theory and Pattern Recognition Applications*. Springer-Verlag, 1993.

[Rip93b] B.D. Ripley. Neural networks and flexible regression and discrimination. In K.V. Mardia, editor, *Statistics and Images*, pages 1–23. Carfax, Abingdon, 1993.

[Rip93c] B.D. Ripley. Statistical aspects of neural networks. In O.E. Barndorff-Nielsen, J.L. Jensen, and W.S. Kendall, editors, *Chaos and Networks-Statistical and Probabilistic Aspects*, pages 40–123. Chapman & Hall, London, 1993.

[Rip94] B. Ripley. Neural network and related methods for classification. *J. R. Statistic. Soc. B*, 46(3):409–456, 1994.

[RP91] H. Rehkugler and T. Poddig. Künstliche neuronale netze in der finanzanalyse: Eine neue Ära der kursprognose? *Wirtschaftsinformatik*, 33(5):365–474, 1991.

[RZF94] A. Refenes, A. Zapranis, and G. Francis. Stock performance modeling using neural networks: a comparative study with regression models. *neural networks*, 7(2):375–388, 1994.

[Sar94] W. Sarle. Neural networks and statistical models. In *Proceedings of the Nineteenth Annual SAS Users Group International Conference, Cary, NC: SAS Institute*, pages 1538–1550. 1994.

[Sar95] W. Sarle. Stopped training and other remedies for overfitting. In *Proceedings of the 27th Symposium on the Interface*. 1995.

[Sca85] L.E. Scales. *Introduction to non-linear optimization*. MacMillan Publishers, London, 1985.

[Sch90] E. Schoeneburg. Stock price prediction using neural networks: a project report. *Neurocomputing*, 2:17–27, 1990.

[Sep94] P. Sephton. Cointegration tests on MARS. *Computational Economics*, 7(1):23–35, 1994.

[SJW92] W. Schiffmann, M. Joost, and R. Werner. Optimization of the backpropagation algorithm for training multilayer perceptrons. Technical report, University of Koblenz, 1992.

[SP94] R. Sharda and R. Patil. A connectionist approach to time series prediction: an empirical test. In R. Trippi and E. Turban, editors, *Neural networks in finance and investing*, pages 451–464. Probus Publishing, 1994.

[SS94] A. Surkan and J. Singleton. Neural networks for bond rating improved by multiple hidden layers. In R. Trippi and E. Turban, editors, *Neural networks in finance and investing*, pages 275–287. Probus Publishing, 1994.

[Sto74] M. Stone. Cross-validatory choice and asssessment of statistical predictions. *Journal of the Royal Statistical Society*, B36:111–147, 1974.

[TAF91] Z. Tang, C. de Almeida, and P. Fishwick. Time series forecasing using neural networks vs. box-jenkins methodology. *Simulation*, 57:303–310, 1991.

[TG91] P. Treleaven and S. Goonatilake. Intelligent financial technologies. In D. Wuertz and F. Murtagh, editors, *Workshop proceedings PASE 1991*, pages 7–26. 1991.

[Tho93a] H. Thodberg. Ace of Bayes: application of neural networks with pruning. Technical Report 1132E, The Danish Meat Research Institute, May 1993.

[Tho93b] R.L. Thomas. *Introductory Econometrics: Theory and Applications*. Longman, London, 1993.

[TK92] K. Tam and M. Kiang. Managerial applications of neural networks: the case of bank failure predictions. *Management Science*, 38(7):926–947, 1992.

[TT93] R. Trippi and E. Turban. *Neural Networks in Finance and Investing: using artificial intelligence to improve real world performance*. Probus Publishing, Chicago, Illinois, 1993.

[Urb92] P. Urbach. Regression analysis: classical and bayesian. *brit. J. Phil. Sci.*, 43:311–342, 1992.

[VD94] W. Verkooijen and H. Daniels. Connectionist projection pursuit regression. *Computational Economics*, 7:155–161, 1994.

[VD95] W. Verkooijen and H. Daniels. Building error-correction models with neural networks: an application to the Dutch mortgage market. *Economic & Financial Computing*, 5(2), 1995.

[Ver95] W. Verkooijen. A neural network approach to long-run exchange rate prediction. *Computational Economics*, 8(4):1–15, 1995.

[Wal89] K.F. Wallis. Macroeconomic forecasting: a survey. *The Economic Journal*, 99:28–61, 1989.

[WBM91] B. Winer, D. Brown, and K. Michels. *Statistical principles in experimental design*. McGraw-Hill, New York, 1991.

[Wei91] S.M. Weiss. Small sample error rate estimation for k-nn classifiers. *IEEE Trans. Pattern Anal. Machine Intell.*, 13:285–289, 1991.

[WG94] A. Weigend and N. Gershenfield. *Time series prediction: forecasting the future and understanding the past*. Addison-Wesley, Reading, 1994.

[Whi88] H. White. Economic prediction using neural networks: the case of IBM daily stock returns. In *Proceedings IEEE International Conference on Neural Networks Vol.II*, pages 451–458. 1988.

[Whi89a] H. White. An additional hidden unit tests for neglected nonlinearity. In *Proceedings of the international joint conference on neural networks*, pages 451–455. IEEE Press, New York, 1989.

[Whi89b] H. White. Learning in artificial neural networks: a statistical perspective. *Neural Computation*, 1:425–464, 1989.

[Whi92] H. White. *Artificial neural networks: approximation and learning theory*. Blackwell Publishers, Cambridge, MA, 1992.

[WHR90] A. Weigend, A. Huberman, and D. Rumelhart. Predicting the future: a connectionist approach. *International Journal of Neural Systems*, 1(3):193–209, 1990.

[WHR91] A. Weigend, A. Huberman, and D. Rumelhart. Generalization by weight-elimination with application to forecasting. In J.E. Moody and Touretzky D.S, editors, *Advances in neural information processing systems 3*, pages 875–882. Morgan Kaufmann Publishers, San Mateo, CA, 1991.

[Wil85] J. Wilcox. A model of the building society sector. Technical Report 26, Bank of England, August 1985.

[WK91] S. Weiss and C. Kulikowski. *Computer systems that learn*. Morgan Kaufmann, San Mateo, CA, 1991.

[WL94] A. Weigend and B LeBaron. Evaluating neural network predictors by bootstrapping. Technical Report CU-CS-725-94, University of Colorado, May 1994.

[WP95] S. Wei and D. Parsley. Purchasing power disparity during the floating rate period: exchange rate volatility, trade barriers and other culprits. Technical Report 5032, National Bureau of Economic Research, 1995.

[WW79] R.J. Wonnacott and T.H. Wonnacott. *Econometrics*. John Wiley & Sons, New York, 1979.

[WY93] P.H. Westfall and S.S. Young. *Resampling-Based Multiple Testing*. John Wiley & Sons, New York, 1993.

[Zha93] P. Zhang. Model selection via multifold cross validation. *The Annals of Statistics*, 21:299–313, 1993.

Index

Center for Economic Research, Tilburg University, The Netherlands
Dissertation Series

No.	Author	Title
1	P.J.J. Herings	Static and Dynamic Aspects of General Disequilibrium Theory; ISBN 90 5668 001 3
2*	Erwin van der Krabben	Urban Dynamics: A Real Estate Perspective - An institutional analysis of the production of the built environment; ISBN 90 5170 390 2
3	Arjan Lejour	Integrating or Desintegrating Welfare States? - a qualitative study to the consequences of economic integration on social insurance; ISBN 90 5668 003 x
4	Bas J.M. Werker	Statistical Methods in Financial Econometrics; ISBN 90 5668 002 1
5	Rudy Douven	Policy Coordination and Convergence in the EU; ISBN 90 5668 004 8
6	Arie J.T.M. Weeren	Coordination in Hierarchical Control; ISBN 90 5668 006 4
7	Herbert Hamers	Sequencing and Delivery Situations: a Game Theoretic Approach; ISBN 90 5668 005 6
8	Annemarie ter Veer	Strategic Decision Making in Politics; ISBN 90 5668 007 2
9	Zaifu Yang	Simplicial Fixed Point Algorithms and Applications; ISBN 90 5668 008 0
10	William Verkooijen	Neural Networks in Economic Modelling - An Empirical Study; ISBN 90 5668 010 2

* Copies can be ordered from Thesis Publishers, P.O. Box 14791, 1001 LG Amsterdam, The Netherlands, phone + 31 20 6255429; fax: +31 20 6203395; e-mail: thesis@thesis.aps.nl

WILLIAM VERKOOIJEN studied technical mathematics at the Technical University of Eindhoven. In 1991 he graduated in the field of decision sciences. In the period 1991-1995 he performed research towards his PhD at Tilburg University. This research synthesises topics from Artificial Intelligence, Econometrics, and Statistics. Since October 1995, William Verkooijen has a position at the ABN AMRO bank.

This dissertation addresses the statistical aspects of neural networks and their usability for solving problems in economics and finance. Neural networks are discussed in a framework of modelling which is generally accepted in econometrics. Within this framework a neural network is regarded as a statistical technique that implements a model-free regression strategy. Model-free regression seems particularly useful in situations where economic theory cannot provide sensible model specifications. Neural networks are applied in three case studies: modelling house prices; predicting the production of new mortgage loans; and predicting foreign exchange rates. From these case studies is concluded that neural networks are a valuable addition to the econometrician's toolbox, but that they are no panacea.

ISBN 90-5668-00-010-2

CentER Dissertation