Preface

This book gives an introduction to basic neural network architectures and learning rules. Emphasis is placed on the mathematical analysis of these networks, on methods of training them and on their application to practical engineering problems in such areas as nonlinear regression, pattern recognition, signal processing, data mining and control systems.

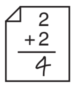
Every effort has been made to present material in a clear and consistent manner so that it can be read and applied with ease. We have included many solved problems to illustrate each topic of discussion. We have also included a number of case studies in the final chapters to demonstrate practical issues that arise when using neural networks on real world problems.

Since this is a book on the design of neural networks, our choice of topics was guided by two principles. First, we wanted to present the most useful and practical neural network architectures, learning rules and training techniques. Second, we wanted the book to be complete in itself and to flow easily from one chapter to the next. For this reason, various introductory materials and chapters on applied mathematics are included just before they are needed for a particular subject. In summary, we have chosen some topics because of their practical importance in the application of neural networks, and other topics because of their importance in explaining how neural networks operate.

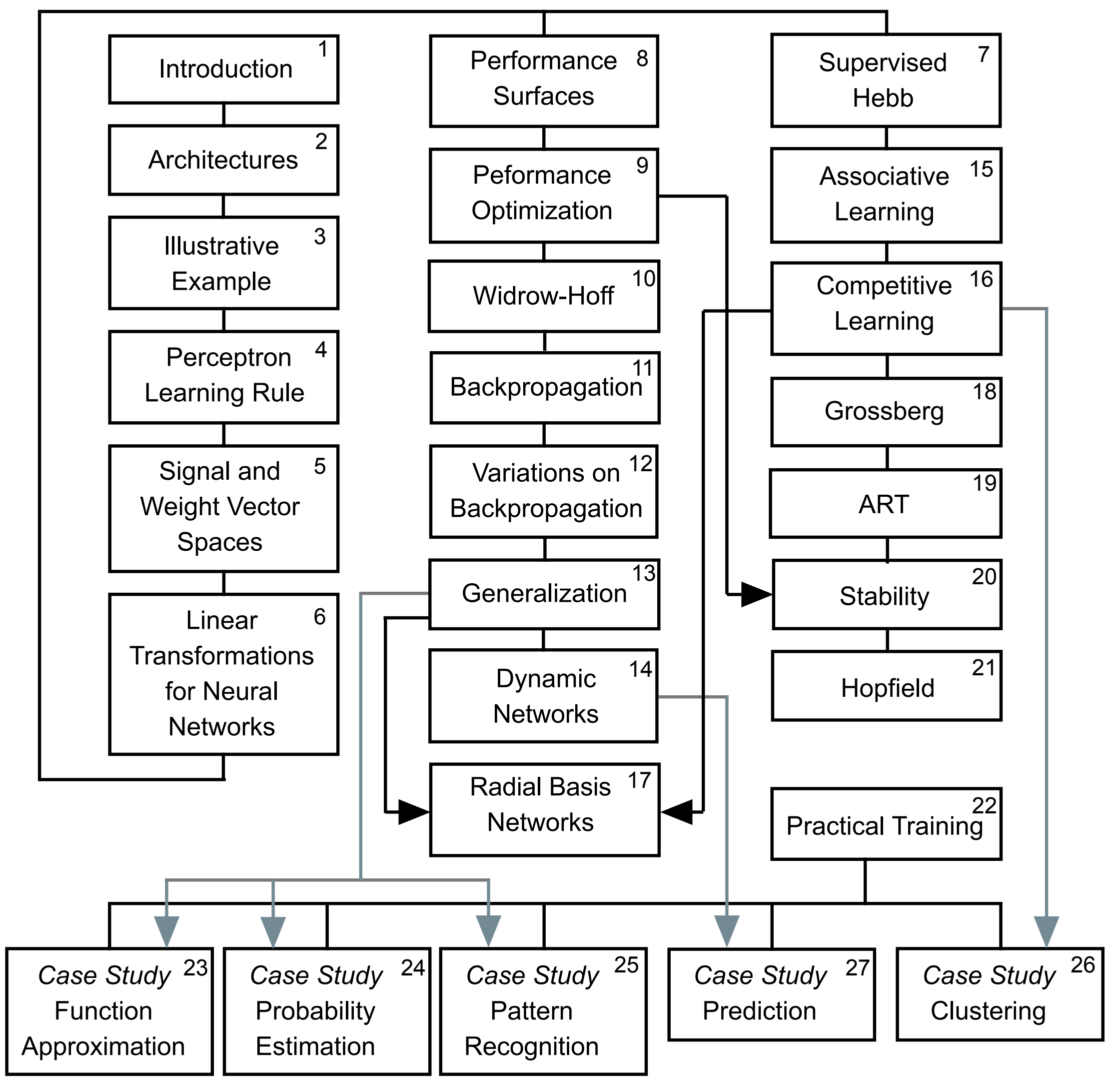
We have omitted many topics that might have been included. We have not, for instance, made this book a catalog or compendium of all known neural network architectures and learning rules, but have instead concentrated on the fundamental concepts. Second, we have not discussed neural network implementation technologies, such as VLSI, optical devices and parallel computers. Finally, we do not present the biological and psychological foundations of neural networks in any depth. These are all important topics, but we hope that we have done the reader a service by focusing on those topics that we consider to be most useful in the design of neural networks and by treating those topics in some depth.

This book has been organized for a one-semester introductory course in neural networks at the senior or first-year graduate level. (It is also suitable for short courses, self-study and reference.) The reader is expected to have some background in linear algebra, probability and differential equations.

Each chapter of the book is divided into the following sections: Objectives, Theory and Examples, Summary of Results, Solved Problems, Epilogue, Further Reading and Exercises. The *Theory and Examples* section comprises the main body of each chapter. It includes the development of fundamental ideas as well as worked examples (indicated by the icon shown here in the left margin). The *Summary of Results* section provides a convenient listing of important equations and concepts and facilitates the use of the book as an industrial reference. About a third of each chapter is devoted to the *Solved Problems* section, which provides detailed examples for all key concepts.



The following figure illustrates the dependencies among the chapters.



Chapters 1 through 6 cover basic concepts that are required for all of the remaining chapters. Chapter 1 is an introduction to the text, with a brief historical background and some basic biology. Chapter 2 describes the basic neural network architectures. The notation that is introduced in this chapter is used throughout the book. In Chapter 3 we present a simple pattern recognition problem and show how it can be solved using three different types of neural networks. These three networks are representative of the types of networks that are presented in the remainder of the text. In addition, the pattern recognition problem presented here provides a common thread of experience throughout the book.

Much of the focus of this book will be on methods for training neural networks to perform various tasks. In Chapter 4 we introduce learning algorithms and present the first practical algorithm: the perceptron learning rule. The perceptron network has fundamental limitations, but it is important for historical reasons and is also a useful tool for introducing key concepts that will be applied to more powerful networks in later chapters.

One of the main objectives of this book is to explain how neural networks operate. For this reason we will weave together neural network topics with important introductory material. For example, linear algebra, which is the core of the mathematics required for understanding neural networks, is reviewed in Chapters 5 and 6. The concepts discussed in these chapters will be used extensively throughout the remainder of the book.

Chapters 7, and 15–19 describe networks and learning rules that are heavily inspired by biology and psychology. They fall into two categories: associative networks and competitive networks. Chapters 7 and 15 introduce basic concepts, while Chapters 16–19 describe more advanced networks.

Chapters 8–14 and 17 develop a class of learning called performance learning, in which a network is trained to optimize its performance. Chapters 8 and 9 introduce the basic concepts of performance learning. Chapters 10–13 apply these concepts to feedforward neural networks of increasing power and complexity, Chapter 14 applies them to dynamic networks and Chapter 17 applies them to radial basis networks, which also use concepts from competitive learning.

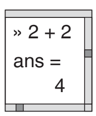
Chapters 20 and 21 discuss recurrent associative memory networks. These networks, which have feedback connections, are dynamical systems. Chapter 20 investigates the stability of these systems. Chapter 21 presents the Hopfield network, which has been one of the most influential recurrent networks.

Chapters 22–27 are different than the preceding chapters. Previous chapters focus on the fundamentals of each type of network and their learning rules. The focus is on understanding the key concepts. In Chapters 22–27, we discuss some practical issues in applying neural networks to real world problems. Chapter 22 describes many practical training tips, and Chapters 23–27 present a series of case studies, in which neural networks are applied to practical problems in function approximation, probability estimation, pattern recognition, clustering and prediction.

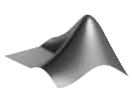
Software

MATLAB is not essential for using this book. The computer exercises can be performed with any available programming language, and the *Neural Network Design Demonstrations*, while helpful, are not critical to understanding the material covered in this book.

However, we have made use of the MATLAB software package to supplement the textbook. This software is widely available and, because of its matrix/vector notation and graphics, is a convenient environment in which to experiment with neural networks. We use MATLAB in two different ways. First, we have included a number of exercises for the reader to perform in MATLAB. Many of the important features of neural networks become apparent only for large-scale problems, which are computationally intensive and not feasible for hand calculations. With MATLAB, neural network algorithms can be quickly implemented, and large-scale problems can be tested conveniently. These MATLAB exercises are identified by the icon shown here to the left. (If MATLAB is not available, any other programming language can be used to perform the exercises.)



The second way in which we use MATLAB is through the *Neural Network Design Demonstrations*, which can be downloaded from the website hagan.okstate.edu/nnd.html. These interactive demonstrations illustrate important concepts in each chapter. After the software has been loaded into the MATLAB directory on your computer (or placed on the MATLAB path), it can be invoked by typing `nnd` at the MATLAB prompt. All demonstrations are easily accessible from a master menu. The icon shown here to the left identifies references to these demonstrations in the text.



The demonstrations require MATLAB or the student edition of MATLAB, version 2010a or later. See Appendix C for specific information on using the demonstration software.

Overheads

As an aid to instructors who are using this text, we have prepared a companion set of overheads. Transparency masters (in Microsoft Powerpoint format or PDF) for each chapter are available on the web at hagan.okstate.edu/nnd.html.

Acknowledgments

We are deeply indebted to the reviewers who have given freely of their time to read all or parts of the drafts of this book and to test various versions of the software. In particular we are most grateful to Professor John Andreae, University of Canterbury; Dan Foresee, AT&T; Dr. Carl Latino, Oklahoma State University; Jack Hagan, MCI; Dr. Gerry Andeen, SRI; and Joan Miller and Margie Jenks, University of Idaho. We also had constructive inputs from our graduate students in ECEN 5733 at Oklahoma State University, ENEL 621 at the University of Canterbury, INSA 0506 at the Institut National des Sciences Appliquées and ECE 5120 at the University of Colorado, who read many drafts, tested the software and provided helpful suggestions for improving the book over the years. We are also grateful to the anonymous reviewers who provided several useful recommendations.

We wish to thank Dr. Peter Gough for inviting us to join the staff in the Electrical and Electronic Engineering Department at the University of Canterbury, Christchurch, New Zealand, and Dr. Andre Titli for inviting us to join the staff at the Laboratoire d'Analyse et d'Architecture des Systèms, Centre National de la Recherche Scientifique, Toulouse, France. Sabbaticals from Oklahoma State University and a year's leave from the University of Idaho gave us the time to write this book. Thanks to Texas Instruments, Halliburton, Cummins, Amgen and NSF, for their support of our neural network research. Thanks to The Mathworks for permission to use material from the *Neural Network Toolbox*.