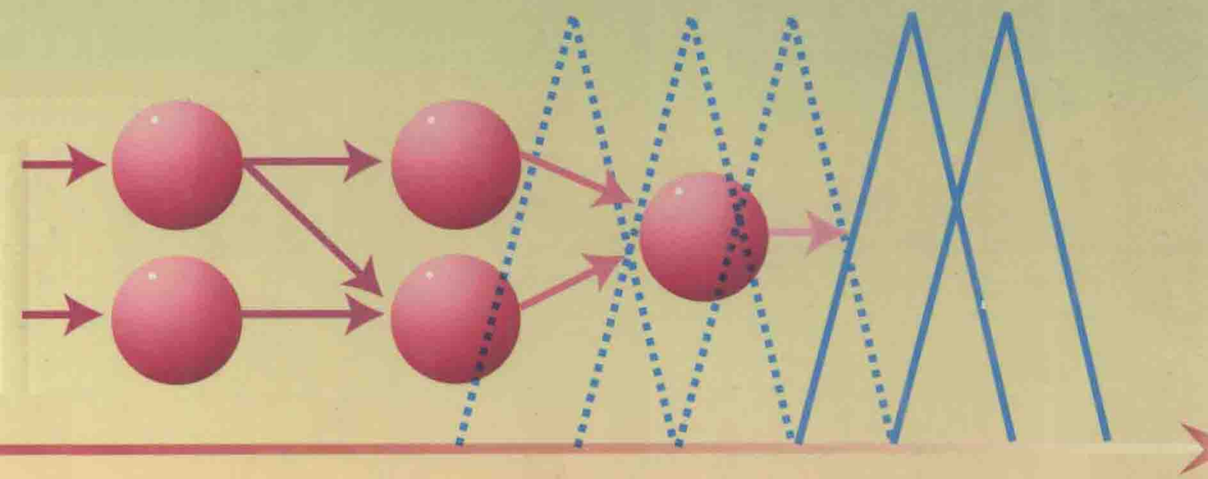

FOUNDATIONS OF

NEURO-FUZZY SYSTEMS

Detlef Nauck

Frank Klawonn

Rudolf Kruse

WILEY

FOUNDATIONS OF

NEURO-FUZZY SYSTEMS

Detlef Nauck

University of Magdeburg, Germany

Frank Klawonn

University of Ostfriesland, Emden, Germany

Rudolf Kruse

University of Magdeburg, Germany

JOHN WILEY & SONS

Chichester • New York • Weinheim • Brisbane • Singapore • Toronto

Originally published in the German language by Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, D-65189 Wiesbaden, Germany, under the title 'Neuronale Netze und Fuzzy Systeme. 2. Auflage (2nd Edition)'.

Baffins Lane, Chichester,
West Sussex PO19 1UD, England
National (01243) 779777
International (+44) 1243 779777
e-mail (for orders and customer service enquiries): cs-books@wiley.co.uk
Visit our Home Page on http://www.wiley.co.uk or http://www.wiley.com

Other Wiley Editorial Offices

John Wiley & Sons, Inc., 605 Third Avenue,
New York, NY 10158-0012, USA

WILEY-VCH Verlag GmbH, Pappelallee 3,
D-69469 Weinheim, Germany

Jacaranda Wiley Ltd, 33 Park Road, Milton,
Queensland 4064, Australia

John Wiley & Sons (Canada) Ltd, 22 Worcester Road,
Rexdale, Ontario M9W 1L1, Canada

John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop #02-01,
Jin Xing Distripark, Singapore 129809

Library of Congress Cataloging-in-Publication Data

Nauck, Detlef.
[Neuronale Netz und Fuzzy Systeme. English]
Foundations of neuro-fuzzy systems / Detlef Nauck, Frank Klawonn, Rudolf Kruse,
p. cm.
Translation of : Neuronale Netze und Fuzzy Systeme. 2nd ed.
Includes bibliographical references and index.
ISBN 0-471-97151-0
1. Neural networks (Computer science) 2. Fuzzy systems.
I. Klawonn, Frank II. Kruse, Rudolf.
QA76.87.N38313 1997
006.3—dc21 97-30674
CIP

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN 0-471-97151-0

Produced from camera-ready copy supplied by the authors
Printed and bound in Great Britain by Bookcraft Ltd, Midsomer Norton
This book is printed on acid-free paper responsibly manufactured from sustainable forestation, for which at least two trees are planted for each one used for paper production.

Preface

Neural networks and fuzzy systems are both very popular techniques in soft computing. The term *soft computing* was coined by Lotfi A. Zadeh, the founder of fuzzy logic. Soft computing includes approaches to human reasoning that try to make use of the human tolerance for incompleteness, uncertainty, imprecision and fuzziness in decision making processes. In addition to neural networks and fuzzy systems it also incorporates evolutionary computation and probabilistic reasoning. Soft computing is especially concerned with combinations of these methodologies. Currently, neuro-fuzzy systems are the most visible approach to such combinations.

Neural networks and their learning capabilities have been examined since the first half of this century. They enjoyed a strong popularity in the 1960s with the development of perceptrons. However, after the limitations of perceptrons became known, interest in neural network research suffered a severe setback. But interest revived when more powerful learning algorithms were discovered in 1985. Since then neural networks have formed a large research area and are used in several applications, from robot control to financial forecasting.

Their most prominent feature is their ability to learn from examples. Using so-called *learning algorithms*, they solve problems by processing a set of training data. Some types of neural networks are comparable to statistical regression or discriminant models. However, they do not explicitly make assumptions on the distribution of their training data, or on the relationship between their input and output variables. From a statistical point of view neural networks are non-parametric models, and for some it can be shown that they are universal function approximators.

There is a drawback of neural networks that can pose a problem for some applications. In general it cannot be proven that a neural network works as expected. Due to its distributed nature the solution that a network has learned cannot be expressed explicitly. A neural network learns, but a user cannot learn from the network. For the user it simply is a black box.

Fuzzy logic was founded by Lotfi A. Zadeh in 1965. It is based on the idea of fuzzy sets that can be used to model *linguistic terms*, i.e. expressions of human language like *large, small, hot, cold*, etc. In fuzzy logic it is possible to formulate fuzzy rules that use such linguistic expressions and apply them to decision making processes. Most applications of fuzzy systems have nothing to do with fuzzy logic in the narrow sense, i.e. they do not consider systems of generalized logical rules. They use fuzzy rules in a broader sense to interpolate functions. Certain types of fuzzy systems are – like neural networks – universal approximators. A popular application of fuzzy systems are fuzzy controllers. Fuzzy systems are, like neural networks, nowadays used for a wide range of applications, from control tasks to data analysis and decision making.

This means fuzzy systems can be used for the same tasks as neural networks. The difference is that fuzzy systems are not created by a learning algorithm. They are built from explicit knowledge which is expressed in the form of linguistic (fuzzy) rules. However, it is sometimes difficult suitably to specify all the parameters of a fuzzy system. If the performance of the fuzzy system is not satisfactory, the parameters must be tuned manually. This tuning process is error-prone and time-consuming.

So the idea of applying some kind of learning algorithm to a fuzzy system is not surprising. From the number of possible ways to accomplish this, the combination with neural network methods was, and still is very popular, and a lot of approaches have been discussed in the literature. The preference for these so-called *neuro-fuzzy systems* is probably due to the fact that neural networks and fuzzy systems – especially fuzzy controllers – became popular roughly at the same time, at the end of the 1980s. Those applying fuzzy controllers, having had trouble tuning them, perhaps have admired the apparent ease with which neural networks learned their parameters. On the other hand neural network users may have admired the transparency and interpretability of a rule-based fuzzy system, while a neural network is only a black box.

In this research monograph we discuss several approaches to neuro-fuzzy systems, including our own models. To keep the book self-contained, we introduce the type of neural networks most often used for combinations with fuzzy systems. We also give an introduction to the basic concepts of fuzzy systems. Readers do not need to have studied neural networks or fuzzy systems to follow this book.

Large parts of this book are based on lectures that were given by the authors at the Universities of Braunschweig, Emden and Magdeburg. Our recent research results on neuro-fuzzy systems are also included. This book is intended for researchers and practitioners in artificial intelligence and for students of computer science or neighbouring areas. The software tools that we present in this book can obtained as public domain software via the Internet at http://fuzzy.cs.uni-magdeburg.de

This book is the translation of our German book that is currently available in its second edition. The text was used in several talks and lectures and has benefited from comments by our students, colleagues and the participants of industrial tutorials. For their support we especially want to thank Juliane Bode, Christian Borgelt, Hermann-Josef Diekgerdes, Patrik Eklund, Wilfried Euing, Jörg Gebhardt, Ingrid Gerdes, Thomas Hoferichter, Volker Leisse, Ulrike Nauck, Andreas Nürnberger, Roland Stellmach, Hideyuki Takagi and Hartmut Wolff. We thank Reinhard ...nbusch from Vieweg Verlag for making this translation possible and we especi... thanks to Roslyn Meredith and Peter Mitchell from Wiley for their support an... patience and Richard Leigh for proofreading the manuscript.

Contents

1

Introduction

At first glance, neural networks and fuzzy systems seem to be totally different area with merely marginal connections to each other. In this introduction we explain briefly the ideas behind these models, and we will show that combinations of neural networks with fuzzy systems, so-called *neural fuzzy* or *neuro-fuzzy systems*, offer a great number of interesting possibilities. In addition, we present an outline of this book.

1.1 Neural Networks

Neural networks, also known as *connectionist models*, are systems that try to make use of some of the known or expected organizing principles of the human brain. They consist of a number of independent, simple processors – the *neurons*. These neurons communicate with each other via weighted connections – the *synaptic weights*. At first, research in this area was driven by neurobiological interests. The modelling of single neurons and the so-called "learning rules" for modifying synaptic weights were the initial research topics.

The famous *perceptron* of Frank Rosenblatt was the first neural model to consider the aspects of processing and storing information. As a "learning machine" it was quite a sensation in 1960 and it started the "golden age" of neural networks. Using a learning rule and a set of examples, the perceptron was able to learn to produce target outputs to given input signals. But soon it became obvious that the perceptron was not able to learn to solve relatively simple problems (e.g. to decide whether there is an even or odd number of ones in an input pattern), and neural network research suffered a severe setback beginning in 1969 – the so called "dark age" of neural networks. Interest and euphoria returned, when new powerful learning algorithms were discovered in 1985. Today [illegible] ural network models are known to be universal approximators, able to [illegible] ly given continuous mapping from inputs to outputs.

[illegible] ern research in neural networks, also called *connectionism*, considers the development of architectures and learning algorithms, and examines the applicability of these models to information processing tasks. The neurobiological plausibility of these "artificial" neural networks usually is only of minor or no importance. But there are many researchers who devote themselves to modelling biological neural networks by artificial neural networks to learn more about the structure of the human brain and the way it works.

We will restrict ourselves to the problem of information processing, and do not consider biological aspects. When we talk about neural networks, we will always refer to artificial neural networks simulated by a computer.

The research on neural networks, their application, and discoveres from neurobiology led to a number of modifications of the older simple neural networks, and to a lot of new models. All these models are based on rather simple processing units or neurons exchanging information via weighted connections. For the sake of a uniform notation, and to stress what the different models have in common and what distinguishes them, we will introduce a *generic model*. It functions as a formal framework for the neural networks and also for the neuro-fuzzy systems discussed in this book.

Different types of neural networks can solve different problems, like pattern recognition, pattern completion, determining similarities between patterns or data – also in terms of interpolation or extrapolation – and automatic classification.

Except in optimization tasks, to which also a number of neural models can be applied, neural networks solve a problem by using a learning procedure that depends on the neural model and the given problem. We distinguish between fixed and free learning tasks, and supervised and unsupervised learning algorithms. A fixed learning task specifies a set of data, where for each given input pattern the target output pattern is known. A neural network is supposed to learn the mapping from input to output space by using a supervised learning algorithm. If the same or similar input patterns are presented to the network after learning, it should produce an appropriate output pattern. Free learning tasks usually define a classification problem, where similarities between patterns have to be found. Each pattern must be assigned to one class without knowing this assignment in advance. It is required that patterns which differ only slightly from each other are put into the same class.

Most of the learning algorithms change the connection weights without modifying the structure of the network, i.e. the number of units and connections. If the learning procedure can find a suitable combination of weights, such that the given problem is solved, then this solution is stored implicitly in the connections of the network. Neural networks are black boxes for their users – it is usually not possible to extract explicit knowledge from them. Neural networks are able to solve difficult problems, but they do not tell us how.

Similar problems arise if we want to use existing knowledge about the connection between input and output patterns. It is not possible to integrate such knowledge into a network to simplify or accelerate the learning process – a neural network always learns from scratch.

The strong points of neural networks are their learning capabilities and their distributed structure that allows for highly parallel software or hardware implementations. The weak points are the inability to integrate knowledge into or to extract it from them, and the determination of their parameters. The number of neurons and their connections, and the parameters of the learning process can usually only be chosen from experience, or they are based on rules of thumb.

1.2 Fuzzy Systems

The last few years have seen increasing interest in fuzzy systems research and applications. This is mainly due to the immense success of Japanese consumer products that make use of fuzzy technology.

The main idea of fuzzy systems is to extend the classical two-valued modelling of concepts and attributes like *tall*, *fast* or *old* in a sense of gradual truth. This means that a person is not just viewed as *tall* or *not tall*, but as *tall to a certain degree* between 0 and 1.

Classical models usually try to avoid *vague*, *imprecise* or *uncertain* information, because it is considered as having a negative influence in an inference process. Fuzzy systems on the other hand deliberately make use of this kind of information. This usually leads to simpler, more suitable models, which are easier to handle and are more familiar to human thinking.

Most applications of fuzzy systems can be found in the area of control engineering – so-called fuzzy control. For this and other kinds of application of fuzzy systems the term "fuzzy logic" is widely used in the literature. To avoid misunderstanding, we distinguish between "fuzzy logic in the narrow sense" and "fuzzy logic in the broad sense". The latter term refers to all models and applications that use fuzzy sets to represent gradual qualities. "Fuzzy logic in the narrow sense" on the other hand only refers to systems that use logical calculi and deduction mechanisms to extend classical two-valued logic to the unit interval as the set of truth values. In Chapter 13 we consider fuzzy logic in the narrow sense, but for the rest of the book we use the term "fuzzy logic" only in the broad sense.

Fuzzy control applications are based on if–then rules. The antecedent of a rule consists of fuzzy descriptions of measured input values, and the consequent defines a – possibly fuzzy – output value for the given input. These rules must not be understood in terms of logical implications, but in terms of a function defined by a number of different samples. Fuzzy control therefore belongs to fuzzy logic in the broad sense, but not to fuzzy logic in the narrow sense.

The determination of concrete membership degrees between 0 and 1 to specify the extent to which an object fulfils a concept, is a general problem in fuzzy systems. For example, if we want to describe the quality *tall* with respect to male adults, we must specify for each height the degree to which it belongs to the concept *tall*. Obviously, this degree must not decline with increasing height. But the exact value given to, say, 182 cm is not obvious.

However, the determination of the membership degrees influences the behaviour of a fuzzy system to a large extent. Especially, if these values are used in chains of calculations or inferences, then a small change in one value may have a great influence on the result. In fuzzy control chaining of results is avoided, and so the influence of single membership values is obvious to some extent. Nevertheless, cumbersome fine tuning of fuzzy controllers by modifying membership degrees is usually necessary to obtain a working system. In complex fuzzy systems manual optimization of membership degrees is virtually impossible.

The benefits of fuzzy systems lie in suitable knowledge representation in the form of if–then rules with fuzzy antecedents and consequents. But problems arise when fuzzy concepts have to be represented by concrete membership degrees, which guarantee that a fuzzy system works as expected. Therefore it is very promising to have learning procedures which can determine these values automatically. The advantages of a combination of fuzzy systems and neural networks are obvious. The drawbacks of both of them – the black box behaviour of neural networks, and the problems of determining concrete membership values for fuzzy systems – could thus be avoided. A combination would constitute an interpretable model which is capable of learning and can use problem-specific prior knowledge.

1.3 Neuro-Fuzzy Systems

The main idea of fuzzy control is to build a model of a human control expert who is capable of controlling a plant without thinking in terms of a mathematical model. The control expert specifies control actions in the form of linguistic rules. These control rules are translated into the framework of fuzzy set theory providing a calculus which can simulate the behaviour of the control expert. The specification of good linguistic rules depends on the knowledge of the control expert. The translation into fuzzy set theory is not formalized and arbitrary choices concerning for example the shape of membership functions can be made. These uncertainties in the process of building a fuzzy controller mostly result in a heuristic tuning process to overcome the initial design errors.

The combination of neural networks and fuzzy controllers into neuro-fuzzy controllers can help to enhance the performance of the controller by using special learning algorithms. The obvious advantages of such a combination were mentioned in the previous section. There are several approaches to neuro-fuzzy systems from which a designer can choose. This choice of a special neuro-fuzzy architecture depends on the available data describing the control task. It is not always possible to provide a *fixed learning task*, i.e. samples of input/output data, so that gradient descent methods like backpropagation can be used for learning. Sometimes there is only a *free learning task*, i.e. just samples of input data, and there is no knowledge about the correct output value (control action). In this case reinforcement learning or clustering methods have to be used. Also the availability of special (neural or fuzzy) hardware has an influence on the choice of a neuro-fuzzy control architecture.

But neuro-fuzzy models are not only applied in control tasks, even if most applications can be found in this domain. There are also approaches in data analysis, where a set of fuzzy rules and fuzzy sets are needed to describe a data set. A possible application is pattern classification, a domain where neural networks are very successful. But a neural network does not provide an explanation for its classification results. A neuro-fuzzy classificator is different: this system learns fuzzy sets and fuzzy rules which on the one hand can perform the desired classification, and on the other hand can also be linguistically interpreted. Thus the result can be checked for plausibility, and prior knowledge can be easily integrated.

In general we will use the term *neuro-fuzzy systems* in this book to refer to all kinds of neuro-fuzzy models regardless of their application area. A neuro-fuzzy system is a combination of neural networks and fuzzy systems in such a way that neural networks, or neural network learning algorithms, are used to determine parameters of fuzzy systems. This means that the main intention of a neuro-fuzzy approach is to create or improve a fuzzy system automatically by means of neural network methods. An even more important aspect is that the system should always be interpretable in terms of fuzzy if–then rules, because it is based on a fuzzy system reflecting vague knowledge.

On the other hand a *fuzzy neural network* is a neural network that uses fuzzy methods to learn faster or to perform better. In this case the improvement of a neural network is the main intention. An interpretation in terms of fuzzy rules is neither important nor possible here, because the system is based on a neural network with black box characteristics. We will not use the term *fuzzy-neuro system*, because it is not clear which of both approaches it refers to.

In this book we use the following taxonomy to describe different combinations of neural networks and fuzzy systems:

- **Fuzzy neural networks:** Fuzzy methods are used to enhance the learning capabilities or the performance of a neural network. This can be done, by using fuzzy rules to change the learning rate [Halgamuge et al., 1994] or by creating a network that works with fuzzy inputs [Narazaki and Ralescu, 1991, Ishibuchi et al., 1995]. These approaches are not to be confused with neuro-fuzzy approaches.
- **Concurrent "neural/fuzzy systems":** A neural network, and a fuzzy system work together on the same task, but without influencing each other, i.e. neither system is used to determine the parameters of the other. Usually the neural network preprocesses the inputs to, or postprocesses the outputs from the fuzzy system. These kinds of models are strictly speaking neither real neuro-fuzzy approaches nor fuzzy neural networks.
- **Cooperative neuro-fuzzy models:** A neural network is used to determine the parameters (rules, rule weights and/or fuzzy sets) of a fuzzy system. After the learning phase, the fuzzy system works without the neural network. These are simple forms of neuro-fuzzy systems, and the simplest form – determining rule weights by neural learning algorithms – is widely used in commercial fuzzy development tools, even though semantical problems can arise, as we show in the following chapters. Cooperative models can be further divided into approaches that: a) learn fuzzy sets offline, b) learn fuzzy rules offline, c) learn fuzzy sets online, d) learn rule weights.
- **Hybrid neuro-fuzzy models:** Modern neuro-fuzzy approaches are of this form. A neural network and a fuzzy system are combined into one homogeneous architecture. The system may be interpreted either as a special neural network with fuzzy parameters, or as a fuzzy system implemented in a parallel distributed form. Some examples of this kind of architecture that are discussed in this book are ANFIS

[Jang, 1993], FuNe [Halgamuge and Glesner, 1993, Halgamuge and Glesner, 1994], Fuzzy Rule Net [Tschichold-Gürman, 1995], GARIC [Berenji and Khedkar, 1992a], NEFCLASS [Nauck and Kruse, 1995], NEFCON [Nauck, 1994a, Nauck and Kruse, 1993, Nauck and Kruse, 1994b], and NEFPROX [Nauck and Kruse, 1997b, Nauck and Kruse, 1997a]. Some of these approaches are reinforcement learning types that are especially suited for control tasks (GARIC, NEFCON), and others are multi-purpose models (ANFIS, FuNe, Fuzzy RuleNet, NEFCLASS, NEFPROX), which use supervised learning, and can be used for data analysis.

1.4 About This Book

The book has 14 chapters. Chapters 2 to 5 briefly review aspects of the most important neural network models and the foundations of fuzzy set theory. We also consider aspects of fuzzy control that play an important role in neuro-fuzzy combinations. The reader is not expected to be familiar with neural networks or fuzzy systems at the outset. The chapters on neural networks and fuzzy systems can be used as an introduction. Readers with knowledge of neural networks and/or fuzzy systems may want to skip Chapters 4 and/or 5. Readers interested in a thorough treatment of neural networks alone should consult one of the numerous textbooks in this field (e.g. [Anderson and Rosenfeld, 1988, Anderson et al., 1990, Aleksander and Morton, 1990, Zurada, 1992, Haykin, 1994, Rojas, 1996]). A detailed introduction to the foundations of fuzzy systems (concerned with fuzzy logic (in the narrow sense), approximate reasoning, possibilistic logic etc.) is given in [Klir and Folger, 1988, Klir and Yuan, 1995, Kruse et al., 1994a, Zimmermann, 1996].

Chapter 2 presents a short account of the historical background to neural networks. In Chapter 3 we explain our generic model of neural networks, and we introduce the terms and notation which we use in the following chapters to describe neural networks and neuro-fuzzy systems.

Chapter 4 is about four important neural network models that are used for neuro-fuzzy combinations. The first model is the *multilayer perceptron*, and its *backpropagation* learning algorithm. The multilayer perceptron is a universal approximator that can approximate any continuous function to any given degree of accuracy, at least in principle. Many neuro-fuzzy combinations use a multilayer perceptron as a basic structure and employ variations of the backpropagation learning procedure. *Radial basis function (RBF) networks* are a similar neural network model. These neural networks have, like the multilayer perceptron, a feedforward architecture. Instead of sigmoids they use RBFs as activation functions in their hidden layer. RBF networks are often used for neuro-fuzzy combinations, because their basis functions can be directly interpreted as fuzzy sets. A neural network model that uses an unsupervised, competitive learning procedure to classify input data is the *self-organizing feature map*. The mapping of input data into classes is not known in advance, and has to be discovered by the network. These kinds of networks can be used to learn fuzzy rules. We also discuss some aspects of neural controllers. They usually learn by reinforcement, a learning strategy that tells the network whether it is behaving correctly or not, but not what the correct output would be. The chapter concludes with a section on the preprocessing of input data.

Chapter 5 provides the necessary background on fuzzy systems, which is needed for the chapters on neuro-fuzzy systems. We explain what fuzzy sets and fuzzy rules are, and how to operate with them. The chapter also discusses two basic types of fuzzy controllers – the Mamdani and the Sugeno type.

The remaining chapters of the book cover the combination of neural networks and fuzzy systems. We show that both approaches essentially solve the same task – the approximation of functions – by using different means. The drawbacks of one approach emerge as the benefits of the other. A combination is therefore extremely interesting.

In Chapter 6 we discuss the modelling of expert behaviour and point out the fundamental possibilities of combining neural networks and fuzzy systems. In Chapters 7 and 8 we present several approaches to neuro-fuzzy systems known from the literature, and discuss their advantages and disadvantages. Chapter 7 covers cooperative systems and Chapter 8 presents several hybrid neuro-fuzzy systems. Most of the models discussed in these two chapters were developed to overcome the problem of manually tuning fuzzy controllers. Some of them are very sophisticated, some of them are simple models with semantical flaws.

As a result of the discussion and comparison of these approaches we present in Chapter 9 a generic model for neuro-fuzzy combinations that we call the *generic fuzzy perceptron*. This model has all the basic features of a neuro-fuzzy system, and can be used to derive models for special domains. A generic model helps to avoid the semantical problems that come with some of the neuro-fuzzy systems discussed in Chapters 7 and 8.

In Chapter 10 we present the NEFCON model that is derived from the generic fuzzy perceptron. NEFCON is a hybrid neuro-fuzzy controller developed by us. This model uses a new kind of learning algorithm based on a *fuzzy error* measure that enables it to learn fuzzy sets and fuzzy rules. After describing the architecture and the learning procedure, we explain the way NEFCON works, and present our software for this model that is available via the Internet.

Many neuro-fuzzy systems can be found in the area of (fuzzy) control engineering. But there are also combinations in other areas. In Chapter 11 we discuss an approach to neuro-fuzzy data analysis where fuzzy rules are used to describe a data set. We present NEFCLASS, our approach to neuro-fuzzy classification that is like NEFCON derived from the fuzzy perceptron. NEFCLASS is also available as public domain software via the Internet.

An approach to neuro-fuzzy function approximation that is also based on the generic fuzzy perceptron is presented in Chapter 12. In Chapter 13 we describe an approach that allows simple PROLOG programs to be interpreted – based on an extension of classical two-valued logic to fuzzy logic – as neural networks. This makes it possible to use a learning procedure for them. We close the book with Chapter 14 where we discuss applications and semantical aspects of neuro-fuzzy systems and provide some guidelines on how to select an appropriate approach.

2

Historical and Biological Aspects

Research on artificial neural networks started around 1940 and was inspired by interest in the neurophysiological fundamentals of the human brain. It was known that the brain consists of interconnected nerve cells – the *neurons* – that influence each other by electrical signals. A neuron conducts its signals via its *axon* that projects from its cell body (*soma*), and it receives signals from other neurons over the connections between their axons and its *dendrites* (see Figure 1). These joints are called *synapses*, and the two connecting cells are separated by a tiny gap about 200 nm wide (*synaptic gap* or *cleft*) [Müller et al., 1995].

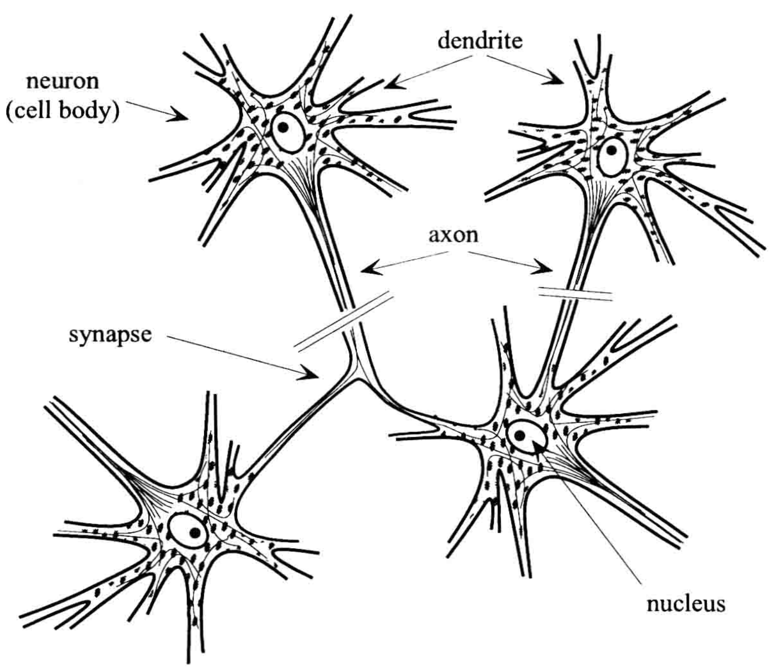

Figure 1 A simplified sketch of four neurons