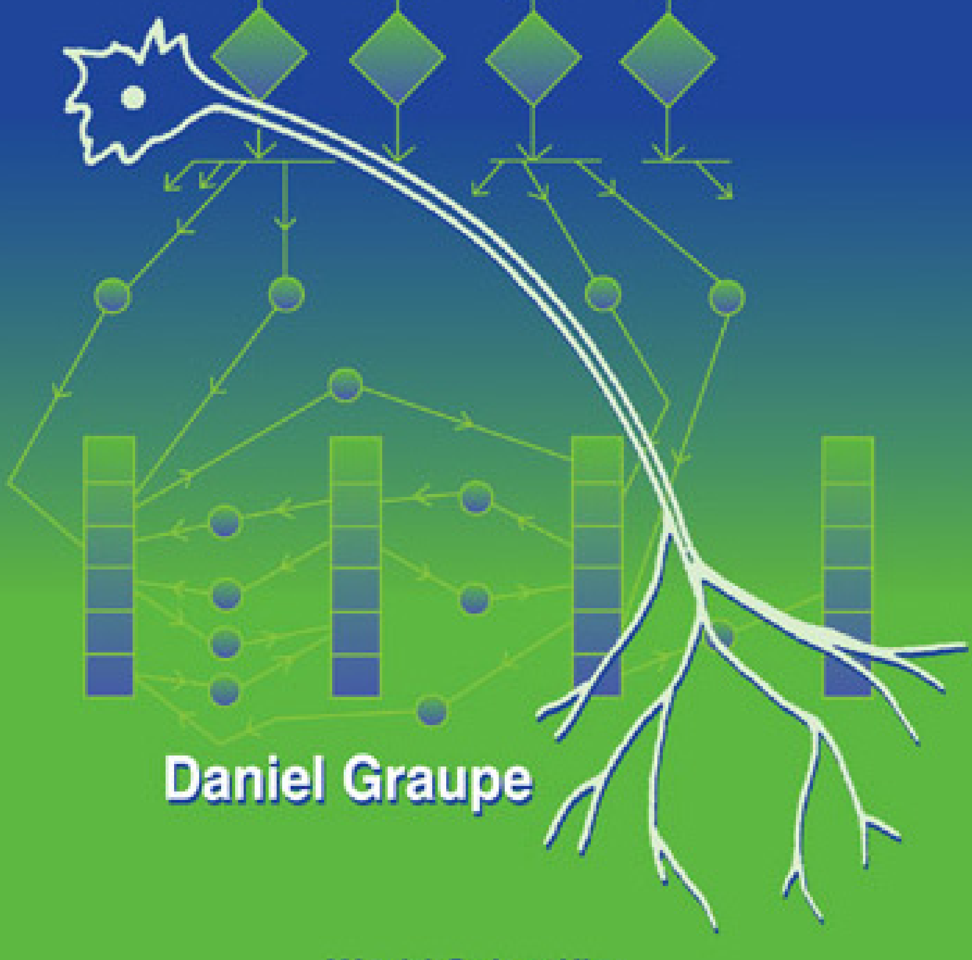
Advanced Series on Circuits and Systems – Vol. 6

PRINCIPLES OF ARTIFICIAL NEURAL NETWORKS

2nd Edition



Daniel Graupe

World Scientific

PRINCIPLES OF ARTIFICIAL NEURAL NETWORKS

2nd Edition

ADVANCED SERIES IN CIRCUITS AND SYSTEMS

Editor-in-Charge: **Wai-Kai Chen** (Univ. Illinois, Chicago, USA)
Associate Editor: **Dieter A. Mlynski** (Univ. Karlsruhe, Germany)

Published

Vol. 1: Interval Methods for Circuit Analysis
by *L. V. Kolev*

Vol. 2: Network Scattering Parameters
by *R. Mavaddat*

Vol. 3: Principles of Artificial Neural Networks
by *D Graupe*

Vol. 4: Computer-Aided Design of Communication Networks
by *Y-S Zhu & W K Chen*

Vol. 5: Feedback Networks: Theory & Circuit Applications
by *J Choma & W K Chen*

Vol. 6: Principles of Artificial Neural Networks (2nd Edition)
by *D Graupe*

Advanced Series on Circuits and Systems – Vol. 6

PRINCIPLES OF ARTIFICIAL NEURAL NETWORKS

2nd Edition

Daniel Graupe

University of Illinois, Chicago, USA

World Scientific

NEW JERSEY • LONDON • SINGAPORE • BEIJING • SHANGHAI • HONG KONG • TAIPEI • CHENNAI

Published by

World Scientific Publishing Co. Pte. Ltd.
5 Toh Tuck Link, Singapore 596224
USA office: 27 Warren Street, Suite 401-402, Hackensack, NJ 07601
UK office: 57 Shelton Street, Covent Garden, London WC2H 9HE

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

PRINCIPLES OF ARTIFICIAL NEURAL NETWORKS (2nd Edition)
Advanced Series on Circuits and Systems – Vol. 6

ISBN-13 978-981-270-624-9
ISBN-10 981-270-624-0

Printed in Singapore.

Dedicated to the memory of my parents,
to my wife Dalia,
to our children, our daughters-in-law and our grandchildren
It is also dedicated to the memory of Dr. Kate H Kohn

This page intentionally left blank

Acknowledgments

I am most thankful to Hubert Kordylewski of the Department of Electrical Engineering and Computer Science of the University of Illinois at Chicago for his help towards the development of LAMSTAR network of Chapter 13 of this text. I am grateful to several students who attended my classes on Neural Network at the Department of Electrical Engineering and Computer Science of the University of Illinois at Chicago over the past fourteen years and who allowed me to append programs they wrote as part of homework assignments and course projects to various chapters of this book. They are Vasanth Arunachalam, Sang Lee, Maxim Kolesnikov, Hubert Kordylewski, Maha Nujeimo, Michele Panzeri, Padmagandha Sahoo, Daniele Scarpazza, Sanjeeb Shah and Yunde Zhong.

I am deeply indebted to the memory of Dr. Kate H. Kohn of Michael Reese Hospital, Chicago and of the College of Medicine of the University of Illinois at Chicago and to Dr. Boris Vern of the College of Medicine of the University of Illinois at Chicago for reviewing parts of the manuscript of this text and for their helpful comments.

Ms. Barbara Aman and the production and editorial staff at World Scientific Publishing Company in Singapore were extremely helpful and patient with me during all phases of preparing this book for print.

This page intentionally left blank

Preface to the First Edition

This book evolved from the lecture notes of a first-year graduate course entitled "Neural Networks" which I taught at the Department of Electrical Engineering and Computer Science of the University of Illinois at Chicago over the years 1990–1996. Whereas that course was a first-year graduate course, several Senior-Year undergraduate students from different engineering departments, attended it with little difficulty. It was mainly for historical and scheduling reasons that the course was a graduate course, since no such course existed in our program of studies and in the curricula of most U.S. universities in the Senior Year Undergraduate program. I therefore consider this book, which closely follows these lecture notes, to be suitable for such undergraduate students. Furthermore, it should be applicable to students at that level from essentially every science and engineering University department. Its prerequisites are the mathematical fundamentals in terms of some linear algebra and calculus, and computational programming skills (not limited to a particular programming language) that all such students possess.

Indeed, I strongly believe that Neural Networks are a field of both intellectual interest and practical value to all such students and young professionals. Artificial neural networks not only provide an understanding into an important computational architecture and methodology, but they also provide an understanding (very simplified, of course) of the mechanism of the biological neural network.

Neural networks were until recently considered as a "toy" by many computer engineers and business executives. This was probably somewhat justified in the past, since neural nets could at best apply to small memories that were analyzable just as successfully by other computational tools. I believe (and I tried in the later chapters below to give some demonstration to support this belief) that neural networks are indeed a valid, and presently, the only efficient tool, to deal with very large memories.

The beauty of such nets is that they can allow and will in the near-future allow, for instance, a computer user to overcome slight errors in representation, in programming (missing a trivial but essential command such as a period or any other symbol or character) and yet have the computer execute the command. This will obviously require a neural network buffer between the keyboard and the main programs. It should allow browsing through the Internet with both fun and efficiency. Advances in VLSI realizations of neural networks should allow in the coming years many concrete applications in control, communications and medical devices, including in artificial limbs and organs and in neural prostheses, such as neuromuscular stimulation aids in certain paralysis situations.

For me as a teacher, it was remarkable to see how students with no background in signal processing or pattern recognition could easily, a few weeks (10–15 hours) into the course, solve speech recognition, character identification and parameter estimation problems as in the case studies included in the text. Such computational capabilities make it clear to me that the merit in the neural network tool is huge. In any other class, students might need to spend many more hours in performing such tasks and will spend so much more computing time. Note that my students used only PCs for these tasks (for simulating all the networks concerned). Since the building blocks of neural nets are so simple, this becomes possible. And this simplicity is the main feature of neural networks: A house fly does not, to the best of my knowledge, use advanced calculus to recognize a pattern (food, danger), nor does its CNS computer work in picosecond-cycle times. Researches into neural networks try, therefore, to find out why this is so. This leads and led to neural network theory and development, and is the guiding light to be followed in this exciting field.

Daniel Graupe
Chicago, IL
January 1997

Preface to the Second Edition

The Second Edition contains certain changes and additions to the First Edition. Apart from corrections of typos and insertion of minor additional details that I considered to be helpful to the reader, I decided to interchange the order of Chapters 4 and 5 and to rewrite Chapter 13 so as to make it easier to apply the LAMSTAR neural network to practical applications. I also moved the Case Study 6.D to become Case Study 4.A, since it is essentially a Perceptron solution.

I consider the Case Studies important to a reader who wishes to see a concrete application of the neural networks considered in the text, including a complete source code for that particular application with explanations on organizing that application. Therefore, I replaced some of the older Case Studies with new ones with more detail and using most current coding languages (MATLAB, Java, C++). To allow better comparison between the various neural network architectures regarding performance, robustness and programming effort, all Chapters dealing with major networks have a Case Study to solve the same problem, namely, character recognition. Consequently, the Case studies 5.A (previously, 4.A, since the order of these chapters is interchanged), 6.A (previously, 6.C), 7.A, 8.A, have all been replaced with new and more detailed Case Studies, all on character recognition in a 6×6 grid. Case Studies on the same problem have been added to Chapter 9, 12 and 13 as Case Studies 9.A, 12.A and 13.A (the old Case Studies 9.A and 13.A now became 9.B and 13.B). Also, a Case Study 7.B on applying the Hopfield Network to the well known Traveling Salesman Problem (TSP) was added to Chapter 7. Other Case Studies remained as in the First Edition.

I hope that these updates will add to the readers' ability to better understand what Neural Networks can do, how they are applied and what the differences are between the different major architectures. I feel that this and the case studies with their source codes and the respective code-design details will help to fill a gap in the literature available to a graduate student or to an advanced undergraduate Senior who is interested to study artificial neural networks or to apply them.

Above all, the text should enable the reader to grasp the very broad range of problems to which neural networks are applicable, especially those that defy analysis and/or are very complex, such as in medicine or finance. It (and its Case Studies) should also help the reader to understand that this is both doable and rather easily programmable and executable.

Daniel Graupe
Chicago, IL
September 2006

Contents

This page intentionally left blank

Chapter 1

Introduction and Role of Artificial Neural Networks

Artificial neural networks are, as their name indicates, computational networks which attempt to simulate, in a gross manner, the networks of nerve cell (neurons) of the biological (human or animal) central nervous system. This simulation is a gross cell-by-cell (neuron-by-neuron, element-by-element) simulation. It borrows from the neurophysiological knowledge of biological neurons and of networks of such biological neurons. It thus differs from conventional (digital or analog) computing machines that serve to replace, enhance or speed-up human brain computation without regard to organization of the computing elements and of their networking. Still, we emphasize that the simulation afforded by neural networks is very gross.

Why then should we view artificial neural networks (denoted below as neural networks or ANNs) as more than an exercise in simulation? We must ask this question especially since, computationally (at least), a conventional digital computer can do everything that an artificial neural network can do.

The answer lies in two aspects of major importance. The neural network, by its simulating a biological neural network, is in fact a novel computer architecture *and* a novel algorithmization architecture relative to conventional computers. *It allows using very simple computational operations* (additions, multiplication and fundamental logic elements) to solve complex, mathematically ill-defined problems, nonlinear problems or stochastic problems. A conventional algorithm will employ complex sets of equations, and will apply to only a given problem and exactly to it. *The ANN will be* (a) *computationally* and algorithmically *very simple* and (b) it will have a *self-organizing feature* to allow it to hold for a wide range of problems.

For example, if a house fly avoids an obstacle or if a mouse avoids a cat, it certainly solves no differential equations on trajectories, nor does it employ complex pattern recognition algorithms. Its brain is very simple, yet it employs a few basic neuronal cells that fundamentally obey the structure of such cells in advanced animals and in man. The artificial neural network's solution will also aim at such (most likely not the same) simplicity. Albert Einstein stated that a solution or a model must be as simple as possible to fit the problem at hand. Biological systems, in order to be as efficient and as versatile as they certainly are despite their inherent slowness (their basic computational step takes about a millisecond versus less than a nanosecond in today's electronic computers), can only do so by converging to the simplest algorithmic architecture that is possible. Whereas high level mathematics and logic can yield a broad general frame for solutions and can be reduced to specific but complicated algorithmization, the neural network's design aims at utmost simplicity and utmost self-organization. A very simple base algorithmic structure lies behind a neural network, but it is one which is highly adaptable to a broad range of problems. We note that at the present state of neural networks their range of adaptability is limited. However, their design is guided to achieve this simplicity and self-organization by its gross simulation of the biological network that is (must be) guided by the same principles.

Another aspect of ANNs that is different and advantageous to conventional computers, at least potentially, is in *its high parallelity* (element-wise parallelity). A conventional digital computer is a *sequential machine*. If one transistor (out of many millions) fails, then the whole machine comes to a halt. In the adult human central nervous system, neurons in the thousands die out each year, whereas brain function is totally unaffected, except when cells at very few key locations should die and this in very large numbers (e.g., major strokes). This insensitivity to damage of few cells is due to the high parallelity of biological neural networks, in contrast to the said sequential design of conventional digital computers (or analog computers, in case of damage to a single operational amplifier or disconnections of a resistor or wire). The same redundancy feature applies to ANNs. However, since presently most ANNs are still simulated on conventional digital computers, this aspect of insensitivity to component failure does not hold. Still, there is an increased availability of ANN hardware in terms of integrated circuits consisting of hundreds and even thousands of ANN neurons on a single chip does hold. [cf. Jabri *et al.*, 1996, Hammerstom, 1990, Haykin, 1994]. In that case, the latter feature of ANNs.

In summary, the excitement in ANNs should not be limited to its greater resemblance to the human brain. Even its degree of self-organizing capability can be built into conventional digital computers using complicated artificial intelligence algorithms. The main contribution of ANNs is that, in its gross imitation of the biological neural network, it allows for very low level programming to allow solving complex problems, especially those that are non-analytical and/or nonlinear and/or nonstationary and/or stochastic, *and* to do so in a self-organizing manner that applies to a wide range of problems with no re-programming or other interference in the program itself. The insensitivity to partial hardware failure is another great attraction, but only when dedicated ANN hardware is used.

It is becoming widely accepted that the advent of ANN will open *new understanding into how* to *simplify* programming and algorithm design for a given end and for a wide range of ends. It should bring attention to the simplest algorithm *without*, of course, *dethroning advanced mathematics* and logic, whose role will always be supreme in mathematical understanding and which will always provide a systematic basis for eventual reduction to specifics.

What is always amazing to many students and to myself is that after six weeks of class, first year engineering graduate students of widely varying backgrounds with no prior background in neural networks or in signal processing or pattern recognition, were able to solve, individually and unassisted, problems of speech recognition, of pattern recognition and character recognition, which could adapt in seconds or in minutes to changes (with a range) in pronunciation or in pattern. They would, by the end of the one-semester course, all be able to demonstrate these programs running and adapting to such changes, using PC simulations of their respective ANNs. My experience is that the study time and the background to achieve the same results by conventional methods by far exceeds that achieved with ANNs.

This, to me, demonstrates the degree of simplicity and generality afforded by ANN; and therefore the potential of ANNs.

Obviously, if one is to solve a set of differential equations, one would not use an ANN, just as one will not ask the mouse or the cat to solve it. But problems of recognition, filtering and control would be problems suited for ANNs. As always, no tool or discipline can be expected to do it all. And then, ANNs are certainly at their infancy. They started in the 1950s; and widespread interest in them dates from the early 1980s. So, all in all, ANNs deserve our serious attention. The days when they were brushed off as a gimmick or as a mere mental exercise are certainly over. Hybrid ANN/serial computer designs should also be considered to utilize the advantages of both designs where appropriate.

This page intentionally left blank

Chapter 2

Fundamentals of Biological Neural Networks

The biological neural network consists of nerve cells (neurons) as in Fig. 2.1, which are interconnected as in Fig. 2.2. The cell body of the neuron, which includes the neuron's nucleus is where most of the neural "computation" takes place. Neural

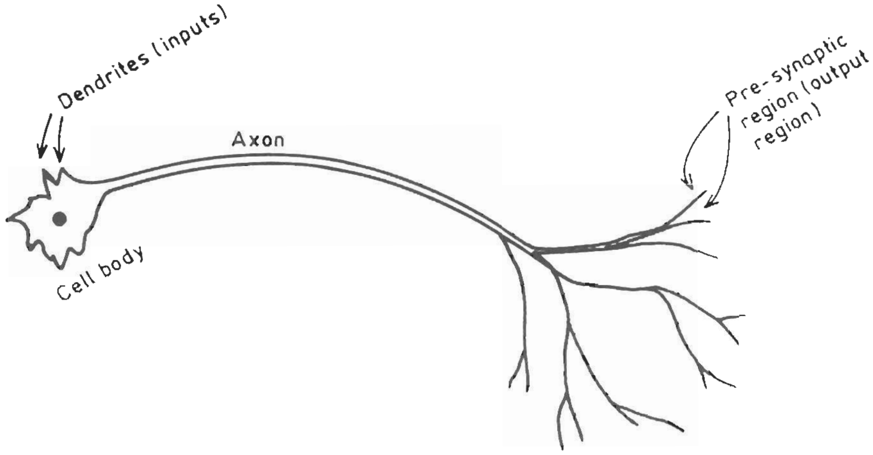


Fig. 2.1. A biological neural cell (neuron).

activity passes from one neuron to another in terms of electrical triggers which travel from one cell to the other down the neuron's axon, by means of an electrochemical process of voltage-gated ion exchange along the axon and of diffusion of neurotransmitter molecules through the membrane over the synaptic gap (Fig. 2.3). The axon can be viewed as a connection wire. However, the mechanism of signal flow is not via electrical conduction but via charge exchange that is transported by diffusion of ions. This transportation process moves along the neuron's cell, down the axon and then through synaptic junctions at the end of the axon via a very narrow synaptic space to the dendrites and/or soma of the next neuron at an average rate of 3 m/sec., as in Fig. 2.3.

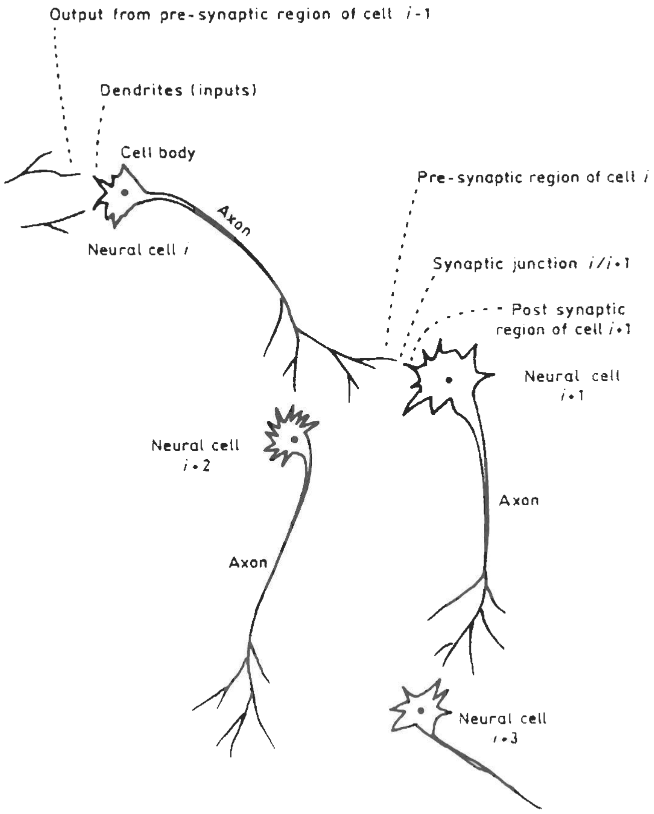


Fig. 2.2. Interconnection of biological neural nets.

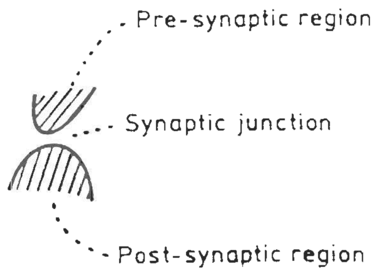


Fig. 2.3. Synaptic junction — detail (of Fig. 2.2).

Figures 2.1 and 2.2 indicate that since a given neuron may have several (hundreds of) synapses, a neuron can connect (pass its message/signal) to many (hundreds of) other neurons. Similarly, since there are many dendrites per each neuron, a single neuron can receive messages (neural signals) from many other neurons. In this manner, the biological neural network interconnects [Ganong, 1973].

It is important to note that not all interconnections, are equally weighted. Some have a higher priority (a higher weight) than others. Also some are excitory and some are inhibitory (serving to block transmission of a message). These differences are effected by differences in chemistry and by the existence of chemical transmitter and modulating substances inside and near the neurons, the axons and in the synaptic junction. This nature of interconnection between neurons and weighting of messages is also fundamental to artificial neural networks (ANNs).

A simple analog of the neural element of Fig. 2.1 is as in Fig. 2.4. In that analog, which is the common building block (neuron) of every artificial neural network, we observe the differences in weighting of messages at the various interconnections (synapses) as mentioned above. Analogs of cell body, dendrite, axon and synaptic junction of the biological neuron of Fig. 2.1 are indicated in the appropriate parts of Fig. 2.4. The biological network of Fig. 2.2 thus becomes the network of Fig. 2.5.

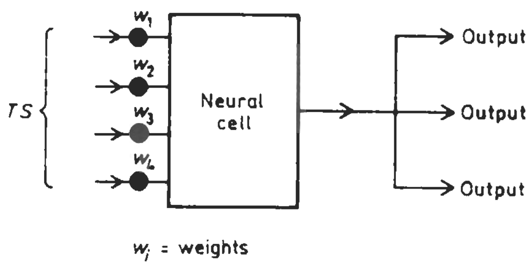


Fig. 2.4. Schematic analog of a biological neural cell.

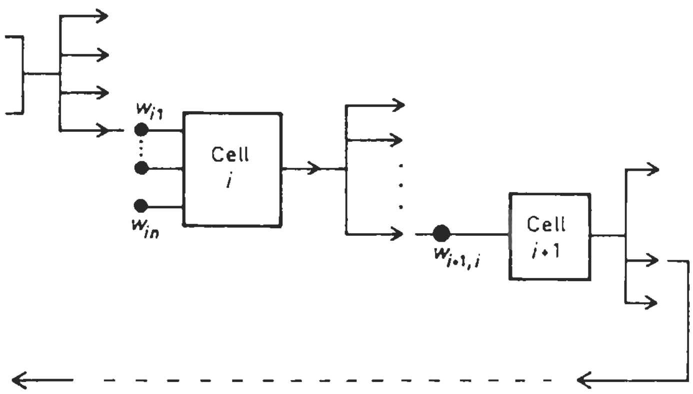


Fig. 2.5. Schematic analog of a biological neural network.