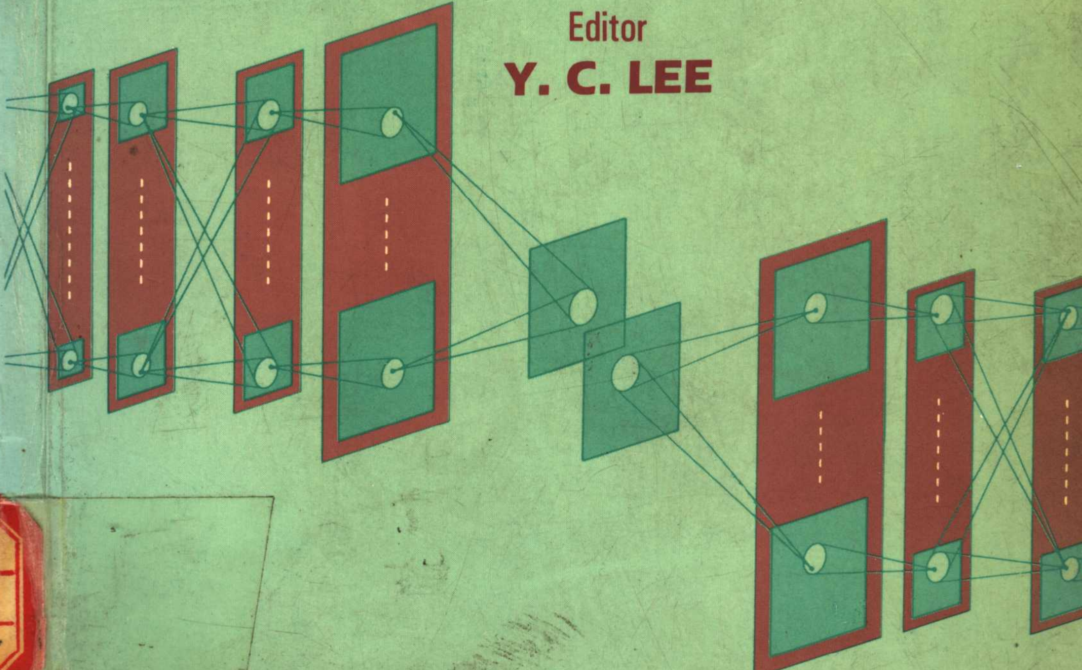
EVOLUTION, LEARNING AND COGNITION

Editor

Y. C. LEE



World Scientific

EVOLUTION, LEARNING AND COGNITION

Editor
Y. C. LEE
Los Alamos National Laboratory

World Scientific
Singapore • New Jersey • London • Hong Kong

Published by

World Scientific Publishing Co. Pte. Ltd.
P O Box 128, Farrer Road, Singapore 9128

USA office: World Scientific Publishing Co., Inc.
687 Hartwell Street, Teaneck, NJ 07666, USA

UK office: World Scientific Publishing Co. Pte. Ltd.
73 Lynton Mead, Totteridge, London N20 8DH, England

Library of Congress Cataloging-in-Publication Data

Evolution, learning & cognition.
1. Neural computers. 2. Artificial intelligence. 3. Cognitive science.
I. Lee, Y. C. (Yee Chun)
QA76.5.E944 1988 006.3 88-33806
ISBN 9971-50-529-0
ISBN 9971-50-530-4 (pbk.)

Printed in Singapore by Utopia Press.

PREFACE

Significant progress has been made in such diverse areas as neural networks, classifier systems, adaptive signal processing, and nonlinear prediction theory in recent years, all of which pertain to a common goal of understanding the adaptive and computational capabilities of natural and artificial complex intelligent systems. The main driving forces behind the current resurgence of interest in computational adaptation and self-organizational techniques are the ready accessibility of inexpensive, fast massively parallel computing devices which permits the modeling of large scale neural and other adaptive networks suitable for practical real world applications, and the realization, by researchers in artificial intelligent systems, of the need to incorporate automatic learning capability into knowledge-based systems in order to deal with the inherent imprecise, incomplete, and ever-changing nature of the real world knowledge base.

The publication of this volume reflects the urgent need for a global overview of this emerging interdisciplinary science. The extraordinarily rapid growth of research effort in the areas of neural networks and genetic algorithm in particular, with the attendant proliferation of research papers and conference proceedings has increasingly forced the researchers to specialize in narrow domains. The speed and magnitude of private companies jumping the bandwagon in their attempt to capitalize on the potential of these powerful techniques for commercial applications have helped to generate the recent wave of public awareness in this new discipline, but at the same time, also have helped to create on media hype surroundings the promise of the new techniques that sometimes laymen and experts alike "find it difficult" to judge whether real progress has been made by reading the frequent press releases and conference papers.

To further compound the problem, along with the big explosion of the R&D effort in both academia and the industrial and commercial sector, came a minor explosion of a confusing array of new products and terminologies. In addition, by and large, this young discipline of learning intelligent systems can still be regarded as a hacker's paradise, with a hodge-podge list of algorithms and tricks, the majorities of which are empirical and were developed with specific applications in mind. As the technology slowly inches toward maturity, it becomes increasingly important to provide a comprehensive yet coherent treatment of the field.

A major issue in the practical application of the new self-organization techniques is the speed at which the intelligent systems can be trained for each task. The learning speed not only depends on the specific system architecture, the learning algorithm employed, but is also strongly dependent on the particular task at hand which defines the shape of the error (or objective) surface. Since the adaptation process of the intelligent system can be essentially described as a kind of optimization process which seeks to improve the performance (and hence reducing the error rate) of the system with each new observation, it is reasonable to expect that the complexity of the landscape (of the error surface) is a direct reflection of the computational complexity of the task given.

The majority of the intelligent problems the systems are expected to solve is most likely to be of the NP-complete type, or at the very least, to lack efficient deterministic algorithm. For those tasks, the way learning complexities scales as the task size is of great concern. However, the potential complexities of the error landscape cannot shoulder all the blame for the slowness of the present learning algorithms. Even tasks which are simple from the algorithmic point of view oftentimes require unacceptably long training sequences. The three dominant learning strategies, i.e., the correlational (or Hebbian) learning, the gradient descent learning, and the Darwinian strategy of random mutation and crossover, are simply not sufficiently "guided" to solve the "ravine tracking" problem due to large eigenvalue spread of the Hessian (i.e., second order) matrix which frequently occurs for large dimensional tasks, even though the mathematical problems associated with the tasks are essentially linear (and therefore simple) in nature.

Another major theoretical issue is the expressive power of the self-modifying intelligent systems. One of the early attempts to build a learning machine was the "perceptron" machine popularized by Rosenblatt. Unfortunately the expressive power of the perceptron was put into question by Minsky and Papert and was found to be inadequate even for certain classes of "easy" problems. Modern neural net architectures are vastly more powerful than their perceptron predecessors. Similarly, the nonneural adaptive mapping network architectures presently being investigated are capable of approximating a large class of smooth and/or hierarchical mappings. Even more impressive are the classifier systems of Holland, described in this volume, since the expressive power of such systems is fully equivalent to that of a Turing machine. However, there seems to be a trade-off between expressive power and the speed of adaptation, as the more expressive systems tend to have more complicated architectures.

Perhaps the least understood aspect of the learning systems is the capabilities of the systems to generalize from learned examples. Some of the generalization capabilities of the adaptive mapping networks such as the locally linear and higher order mappers of Farmer (this volume), the default hierarchy net (also this volume), and the single layer perceptron can be attributed either to either the smoothness hypothesis assumed by these systems which allows explicit interpolation algorithms to be used for generalization, or to the built-in hierarchical memory organization and the sequential learning algorithm which favors generalization through hierarchy formation. The generalization that seems to be provided by classifier systems and neural nets with hidden processing units is much more difficult to comprehend.

For some investigators and industrial users, the hidden neural nets and classifier systems hold a forbidding aura of deep mystery. A few people even have gone so far as to claim this to be a major virtue of the systems and evidence of the supposed superiority of the so-called "brain metaphor". While this assertion may delight neural modelers and thrill the public media, it does nothing to clarify the matter. There is simply no substitute for a sound mathematical investigation of the characteristics of neural generalization, even if conducted in relatively circumscribed domains.

In order to partially address each of the above issues, we invited active researchers who are leaders and pioneers in their respective fields to contribute to this volume. Even though numerous research papers have already appeared in widely disparate forums, with the bulk being in the form of conference and workshop proceedings, there has been no single volume which provides access to state-of-the-art research in the broad discipline of self-organizing intelligent systems. We have tried to include the applications of one methodology in several, different domains as well as the applications of distinct methodologies to the same problems so long as it is feasible. This should allow the comparison of different methodologies and hence promote cross-fertilization.

The book oan be divided roughly into three almost equal parts. In the first part, the papers selected are of a more general, mathematical nature, and as such, they provide a mathematical introduction to the general subject. The second part of the book comprises description and formulation of various adaptive architectures, whereas the last part of the book is mostly devoted to applications. Such division, however, is only approximate, since all papers selected for this volume are essentially self-contained, each with its own architectural description, mathematical formulation, and application results or suggestions.

Throughout this volume, the aim was to provide the most up-to-date account of the present status in learning intelligent system methodologies in diverse areas, and to suggest directions for future research. If this book can convey the excitement experienced by those active in this new discipline and can provide stimulus to beginning readers to participate in the advancement of the subject, then the purpose of this volume will be amply served.

The author would like to thank Dr. David K. Campbell, Director of the Center for Nonlinear Studies at Los Alamos National Laboratory, for suggesting this project to me and for his continuous support and encouragment. I also wish to thank the editorial staff of World Scientific for their valuable expert technical help.

Los Alamos
1988

Y. C. Lee

CONTENTS

PART ONE: MATHEMATICAL THEORY

PART TWO: ARCHITECTURAL DESIGN

PART THREE: APPLICATIONS

Part One
MATHEMATICAL THEORY

Connectionist Learning Through Gradient Following

Ronald J. Williams
College of Computer Science
Northeastern University
Boston, MA 02115

INTRODUCTION

Consider the two questions: (1) What are the processing principles, learned or innate, used by the brain to compute a given sensorimotor or cognitive function, such as visual recognition of one's grandmother, auditory recognition of a familiar melody, motor commands to control the swing of a bat at a 90 m.p.h. fastball, or a decision to make a particular chess move? (2) What are the principles used by the brain to adapt itself to meet the needs of the particular environment it finds itself in at any particular stage of its existence, so that, for example, it can improve at any of the above tasks?

One may seek answers to these questions for their own sake or as a means of identifying techniques for use in artificial systems having similar capabilities. Ultimately, the answers to these two difficult questions will depend on empirical studies of the brain itself. In the meantime, however, one can try to approach them by studying simplified formal models. The difficulty, of course, is to decide what constitutes a valid model of processing in the brain, of various sensorimotor and cognitive functions, and of adaptation and learning. Perhaps an even more fundamental difficulty is to resolve the philosophical problem of what constitutes processing *principles*, as opposed to *details*. Because of the wide latitude possible in any of these areas, it is not surprising that a wide variety of approaches to these questions have been investigated by various researchers.

There are those psychologists and artificial intelligence researchers who believe that the principles of brain-like processing are best expressed in the language of computational symbol manipulation (e.g., Newell, 1980)—at least for those specifically high-level cognitive functions, as distinguished from the more low-level sensorimotor functions. At another extreme are neurophysiologists, who would like to explain brain functioning in terms of the biochemical details of synaptic communication and neural cell growth.

Somewhere between these two extremes in the issue of what constitutes processing principles versus mere details is the study of what are variously called

Preparation of this article was supported by the National Science Foundation under grant IRI-8703566.

connectionist systems, parallel distributed processing networks, or *artificial neural systems*. The unifying feature of these systems is that they consist of highly interconnected networks of relatively simple processing units, the computational properties of the system being a result of the collective dynamics of the network. This approach is distinct from that of modeling biological neural networks because the individual processing units are not constrained to match in any but the most superficial way the details of biological neuron functioning. A common disclaimer is to use the term *neuron-like* to describe the individual units and *neurally inspired* to describe the resulting models.

There are many reasons why this approach is considered worthwhile for helping to provide valuable insights into brain functioning as well as suggesting useful approaches to the design of artificial systems having brain-like capabilities. Among the attractive features of such networks are: (1) their high degree of parallelism, with computational processing broadly distributed across possibly very many units; (2) their powerful associative memory properties, including best-match generalization, content addressability, and graceful degradation; and (3) their ability to rapidly compute "near-optimal" solutions to highly constrained optimization problems. These networks can form nonlinear mappings (such as Boolean functions) and are often constructed to manifest interesting nonlinear dynamics. Many of these properties are explored and discussed in, e.g., Hinton and Anderson (1981), Hopfield (1982), Hinton and Sejnowski (1983), Kohonen (1984), Feldman (1985), Hopfield and Tank (1985), Rumelhart and McClelland (1986), and McClelland and Rumelhart (1986).

This article will describe two particular approaches to arriving at answers to the questions posed above which are appropriate to a connectionist view of brain-like processing. In particular, we will examine two classes of learning algorithms for such networks, where the term "learning" is intended to be interpreted quite generally as something that can be applied either on-line, as in its usual sense, or off-line. Thus the learning algorithms to be described here may be thought of as possible answers to the second question posed above, or, alternatively, as automated techniques for proposing candidate answers to the first question.

The learning algorithms considered here are appropriate to two particular formalizations of the learning problem for a connectionist system. While these two paradigms are quite different and make different assumptions about the nature of the computation performed by the units in the net, the common thread is that algorithms for each case can be derived mathematically by first formulating the learning problem as an optimization problem and then using the simple but powerful principle of stochastic hill-climbing in this criterion function. Specifically, algorithms are presented here for each of these learning paradigms which follow the gradient—statistically, at least—of appropriate performance measures and have the further important property of being implementable locally.

CONNECTIONIST SYSTEMS

A connectionist system is simply a network of computational nodes, called *units*, and a collection of one-way signal paths, or *connections*, between them. It is assumed that this network interacts with an environment, so that some of the units, called *input units*, receive signals from the environment, and other units, called *output units*, transmit signals to the environment. In general, there may be units in the network which are neither input nor output units, and these are called *hidden units*. Hidden units provide a particular challenge for certain types of learning task because neither their actual or desired states are specified by the particular task.

There are a variety of assumptions which can be made concerning the nature of the computation performed by the individual units within a connectionist network. Each unit computes an output signal as some function of that unit's several input signals, and these input signals are themselves either equal to the outputs of units in the net or signals received from the environment. In general, these input and output signals are time-varying, but in certain restricted cases it may not be necessary to make this time dependence explicit. Input and output values of units in the net may be assumed to be discrete (Hopfield, 1982; Hinton & Sejnowski, 1983; Rosenblatt, 1962; Barto & Anderson, 1985) or continuous (Hopfield & Tank, 1985; Kohonen, 1984; Widrow & Hoff, 1960; Rumelhart et al., 1986), and the input/output function of units may be assumed to be deterministic (Hopfield, 1982; Hopfield & Tank, 1985; Kohonen, 1984; Rosenblatt, 1962; Widrow & Hoff, 1960; Rumelhart et al., 1986) or stochastic (Hinton & Sejnowski, 1983; Ackley et al., 1985; Barto & Anderson, 1985). In addition, when the time-varying nature of these signals propagating through the net is important, time may be modeled as discrete (Ackley et al., 1985; Barto & Anderson, 1985; Rumelhart et al., 1986) or continuous (Hopfield & Tank, 1985; Kohonen, 1984), with updating of output values performed synchronously (Barto & Anderson, 1985; Rumelhart et al., 1986) or asynchronously (Hopfield, 1982; Hinton & Sejnowski, 1983; Ackley et al., 1985). Still another point of variation is whether the network is assumed to have feedback loops (Hopfield, 1982; Hopfield & Tank, 1985; Hinton & Sejnowski, 1983; Kohonen, 1984, Ackley et al., 1985) or be acyclic (Barto & Anderson, 1985; Rumelhart et al., 1986).

Throughout all these variations is the common pair of assumptions, intended to capture the idea expressed in describing the computation as *neuron-like*, that: (1) signals transmitted along the connections are (time-varying) *scalars*; and (2) the computation performed at each unit is relatively simple. This second assumption is vague, but intended to rule out, for example, sophisticated encoding/decoding schemes as would be used for communication between two digital computing devices. Weighted analog summation combined with some simple nonlinearity is a typical example of a computation which is considered to satisfy this second criterion. Below we will consider some specific examples of computational units for connectionist networks.

LEARNING

There are a number of possible formulations of the learning problem for a connectionist system. The two particular learning paradigms of interest in this article are *supervised learning* and *associative reinforcement learning*, both of which involve learning on the basis of experience with a finite set of examples. The main distinction between these is the nature of the feedback provided to the system in the two cases. Figures 1 and 2 illustrate networks facing the two types of learning problem. For supervised learning the system is presented with the desired output for each training instance, while for reinforcement learning the system produces a response which is then evaluated using a scalar value indicating the appropriateness of the response. The objective in the supervised learning problem is to find network parameters which minimize some measure of the difference between actual and desired response, while the objective in the associative reinforcement learning problem is to find network parameters maximizing some function of the evaluation signal. Since the training examples for supervised learning consist of input/desired-output pairs, supervised learning might also be thought of as storage of such pairs (albeit in a way designed to permit efficient retrieval and generalization).

It is interesting to note that while there is a long history of attempts to develop what have been called self-organizing procedures for connectionist networks, it is only recently that certain obstacles faced by earlier approaches have been satisfactorily overcome. In particular, a major difficulty for the supervised learning problem has been in devising learning algorithms capable of providing effective adjustment of the parameters associated with hidden units in the network. For this reason, earlier research efforts (e.g., Rosenblatt, 1962; Widrow & Hoff, 1960) generally contented themselves with restricting learning in such networks to certain limited portions which excluded the hidden units.

It should be noted that other formulations of the learning problem are possible. One leading competitor in connectionist circles to those discussed here is that of *unsupervised learning*, in which learning occurs in the absence of any performance feedback. In this paradigm, the objective is for the network to discover statistical regularities or *clusters* in the stream of input patterns. Although we do not consider such learning procedures here, it is worth pointing out why such techniques have been (and continue to be) of interest. One reason is that, until fairly recently, there appeared to be no alternative for training the hidden units in multilayer nets in supervised or associative reinforcement learning tasks. By not depending on performance feedback of any sort, such techniques allow the independent self-organization of individual portions (typically single layers) of a network. Of course, there can thus be no assurance that the resulting performance is desirable (much less optimal) for a given task. With the recent development of promising algorithms for supervised and associative reinforcement learning in multilayer networks (Ackley, Hinton, & Sejnowski, 1985; Barto & Anandan, 1985; Rumelhart, Hinton, & Williams, 1986), the importance of this use for unsupervised learning procedures has diminished. Another source of the appeal of such procedures is their simplicity and biological plausibility. Much of the work of Grossberg (e.g., 1976) makes use of this class of algorithm, and Kohonen (1984) has demonstrated some interesting properties of certain algorithms of this type. Discussion of this general approach to learning may be found in Rumelhart and Zipser (1985).

The specific algorithms to be described here together with their gradient-following properties are the *back-propagation* algorithm (Rumelhart, Hinton, & Williams, 1986; Parker, 1982, 1985; Werbos, 1974) for supervised learning in networks of deterministic units and the *REINFORCE* class of algorithms (Williams, 1986, 1987) for associative reinforcement learning in networks of stochastic units. These latter algorithms are closely related to that investigated by Barto and Anderson (1985). Another recently developed stochastic hill-climbing algorithm which will not be discussed here is the *Boltzmann machine* learning algorithm of Ackley, Hinton, & Sejnowski (1985).

Supervised Learning vs. Associative Reinforcement Learning

Since this article discusses two different formulations of the learning problem and describes algorithms for each, it is useful to clarify the distinctions between the two and discuss briefly the question of their appropriateness.

In the associative reinforcement learning paradigm a network and its training environment interact in the following manner: The network receives a time-varying vector of inputs from the environment and sends a time-varying vector of outputs (also called *actions*) to the environment. In addition, it receives a time-varying scalar signal, called *reinforcement*, from the environment. The objective of learning is for the network to try to maximize some function of this reinforcement signal, such as the expectation of its value on the upcoming time step or the expectation of some integral of its values over all future time, as appropriate for the particular task. The precise nature of the computation of reinforcement by the environment can be anything appropriate for the particular problem and is assumed to be unknown to the learning system. In general, it is some function, deterministic or stochastic, of the input patterns produced by the environment and the output patterns it receives from the network. Figure 1 depicts the interaction between a network and its environment in an associative reinforcement learning situation.

This formulation should be contrasted with the *supervised learning* paradigm, in which the network receives a time-varying vector signal, indicating *desired output*, from the environment, rather than the scalar reinforcement signal, and the objective is for the network's output to match the desired output as closely as possible. This distinction is sometimes summarized by saying that the feedback provided to the network is *instructive* in the case of supervised learning and *evaluative* in the case of reinforcement learning. Figure 2 depicts the interaction between a network and its environment in a supervised learning situation.

We do not concern ourselves here with which is the more appropriate formalization in general, but simply note that each seems to have its place. The idea of matching a specified output pattern seems appropriate for certain problems dealing

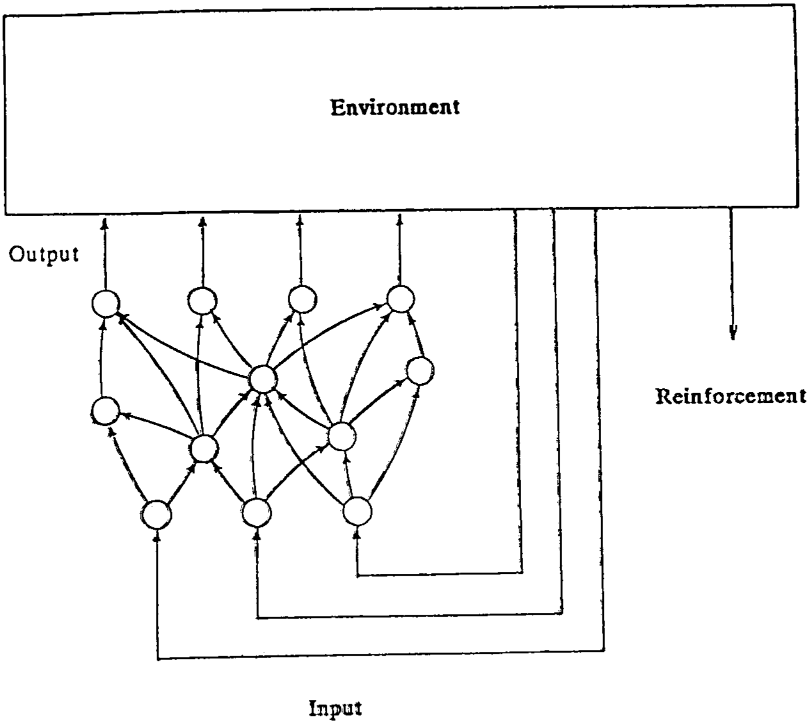


Figure 1. A connectionist network and its training environment for the associative reinforcement learning problem. The precise operation of this system consists of the following four phases:

1. The environment picks an input pattern for the network randomly (the distribution of which is assumed to be independent of prior events within the network/environment system).
2. As the input pattern to each unit becomes available, it computes its output. Thus "activation" passes through the network from the input side to the output side.
3. After all the units at the output side of the network have computed their outputs the environment evaluates the result as a (possibly stochastic) function of the given input and output patterns.
4. Each unit changes its internal parameters according to some specified function of the current value of those parameters, the input it received, the output it produced, and the environment's evaluation. The precise manner in which the evaluation, or *reinforcement*, signal is used by the individual units depends on the learning algorithm to be applied. In the simplest case, the reinforcement signal is simply broadcast to all units, but the use of additional units or interconnections designed to help in the learning process is also possible.