

INFORMATION TO USERS

This reproduction was made from a copy of a manuscript sent to us for publication and microfilming. While the most advanced technology has been used to photograph and reproduce this manuscript, the quality of the reproduction is heavily dependent upon the quality of the material submitted. Pages in any manuscript may have indistinct print. In all cases the best available copy has been filmed.

The following explanation of techniques is provided to help clarify notations which may appear on this reproduction.

1. Manuscripts may not always be complete. When it is not possible to obtain missing pages, a note appears to indicate this.
2. When copyrighted materials are removed from the manuscript, a note appears to indicate this.
3. Oversize materials (maps, drawings, and charts) are photographed by sectioning the original, beginning at the upper left hand corner and continuing from left to right in equal sections with small overlaps. Each oversize page is also filmed as one exposure and is available, for an additional charge, as a standard 35mm slide or in black and white paper format.*
4. Most photographs reproduce acceptably on positive microfilm or microfiche but lack clarity on xerographic copies made from the microfilm. For an additional charge, all photographs are available in black and white standard 35mm slide format.*

*For more information about black and white slides or enlarged paper reproductions, please contact the Dissertations Customer Services Department.

U·M·I Dissertation
Information Service

University Microfilms International
A Bell & Howell Information Company
300 N. Zeeb Road, Ann Arbor, Michigan 48106

PREVIEW

8701132

Anderson, Charles William

LEARNING AND PROBLEM SOLVING WITH MULTILAYER CONNECTIONIST
SYSTEMS

University of Massachusetts

Ph.D. 1986

University
Microfilms
International

300 N. Zeeb Road, Ann Arbor, MI 48106

Copyright 1986

by

Anderson, Charles William

All Rights Reserved

PREVIEW

**LEARNING AND PROBLEM SOLVING
WITH MULTILAYER CONNECTIONIST SYSTEMS**

A Dissertation Presented

By

CHARLES WILLIAM ANDERSON

**Submitted to the Graduate School of the
University of Massachusetts in partial fulfillment
of the requirements for the degree of**

DOCTOR OF PHILOSOPHY

September 1986

Department of Computer and Information Science

© Charles William Anderson 1986
All Rights Reserved

This research was supported by the Air Force Office of Scientific Research and the Avionics Laboratory (Air Force Wright Aeronautical Laboratories) through contracts F33615-77-C-1191, F33615-80-C-1088, and F33615-83-C-1078.

**LEARNING AND PROBLEM SOLVING
WITH MULTILAYER CONNECTIONIST SYSTEMS**

A Dissertation Presented

By

CHARLES WILLIAM ANDERSON

Approved as to style and content by:



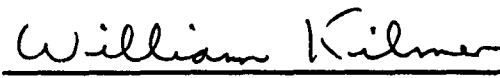
Andrew G. Barto, Chairperson of Committee




Paul E. Utgoff, Member



Michael A. Arbib, Member



William L. Kilmer, Outside Member


Conrad A. Wogrin, Department Chair
Computer and Information Science

**DEDICATED TO
STACEY AND JOSEPH**

ACKNOWLEDGEMENTS

Research in areas where there are many possible paths to follow requires a keen eye for crucial issues. The study of learning systems is such an area. Through the years of working with Andy Barto and Rich Sutton, I have observed many instances of “fluff cutting” and the exposure of basic issues. I thank both Andy and Rich for the insights that have rubbed off on me. I also thank Andy for opening up an infinite world of perspectives on learning, ranging from engineering principles to neural processing theories. I thank Rich for showing me the most important step in doing “science”—simplify your questions by isolating the issues.

Several people contributed to the readability of this dissertation. Andy spent much time carefully reading several drafts. Through his efforts the clarity is much improved. I thank Paul Utgoff, Michael Arbib, and Bill Kilmer for reading drafts of this dissertation and providing valuable criticisms. Paul provided a non-connectionist perspective that widened my view considerably. He never hesitated to work out differences in terms and methodologies that have been developed through research with connectionist vs. symbolic representations. I thank Stephen Judd, Brian Pinette, Jonathan Bachrach, and Robbie Jacobs for commenting on an early draft and for many interesting discussions.

I thank Andy, Rich, Harry Klopf, Michael Arbib, Nico Spinelli, Bill Kilmer, and the AFOSR for starting and maintaining the research project that supported the work reported in this dissertation. I thank Susan Parker for the skill with which she administered the project. And I thank the COINS Department at UMass and the RCF Staff for the maintenance of the research computing environment. Much of the computer graphics software used to generate figures of

this dissertation is based on graphics tools provided by Rich Sutton and Andy Cromarty.

Most importantly, I thank Stacey and Joseph for always being there to lift my spirits while I pursued distant milestones and to share my excitement upon reaching them. Their faith and confidence helped me maintain a proper perspective.

PREVIEW

ABSTRACT

LEARNING AND PROBLEM SOLVING WITH MULTILAYER CONNECTIONIST SYSTEMS

September 1986

Charles William Anderson

B.S., University of Nebraska

M.S., University of Massachusetts

Ph.D., University of Massachusetts

Directed by: Professor Andrew G. Barto

The difficulties of learning in multilayered networks of computational units has limited the use of connectionist systems in complex domains. This dissertation elucidates the issues of learning in a network's *hidden* units, and reviews methods for addressing these issues that have been developed through the years. Issues of learning in hidden units are shown to be analogous to learning issues for multilayer systems employing symbolic representations.

Comparisons of a number of algorithms for learning in hidden units are made by applying them in a consistent manner to several tasks. Recently developed algorithms, including Rumelhart, et al.'s, error back-propagation algorithm and Barto, et al.'s, reinforcement-learning algorithms, learn the solutions to the tasks much more successfully than methods of the past. A novel algorithm is examined

that combines aspects of reinforcement learning and a *data-directed* search for useful weights, and is shown to out perform reinforcement-learning algorithms.

A connectionist framework for the learning of strategies is described which combines the error back-propagation algorithm for learning in hidden units with Sutton's AHC algorithm to learn evaluation functions and with a reinforcement-learning algorithm to learn search heuristics. The generality of this hybrid system is demonstrated through successful applications to a numerical, pole-balancing task and to the Tower of Hanoi puzzle. Features developed by the hidden units in solving these tasks are analyzed. Comparisons with other approaches to each task are made.

TABLE OF CONTENTS

DEDICATIONS	iv
ACKNOWLEDGEMENTS	v
ABSTRACT	vii
LIST OF TABLES	xiii
LIST OF FIGURES	xv
CHAPTER	
I. INTRODUCTION	1
Connectionist Systems	3
Credit Assignment in Multilayer Systems	10
Reinforcement Learning	13
Problem Solving	15
Research Objective and Method	17
Reader's Guide	19
II. NEW FEATURES AND THE FACILITATION OF LEARNING	21
The Problem of Missing Features	22
Missing Terms in LEX	23
The Multiplexer Task	27
The Perceptron Learning Algorithm	29
Performance Measures	30
Original Representation	31
An Ideal Representation	32
A Representation Resulting in No Generalization	33
New Features Added to Original Representation	34
Results	35
New Features That Add Beneficial Generalization	38
The Input-Cluster Task	39
Original Representation	40
New Features—Basis Vectors as Class Labels	40
New Features—Class Labels with Generalization	41
Results	42

New Output Features—Correlations Among Output Components . .	44
The Output-Vector Task	46
Changes to Performance Measures	47
Original Representation	48
New Features That Correlate Output Components	48
Results	50
New Intermediate Concepts in Production Systems	52
Taxonomy Points	54
Switchover Points	55
Summary	56
 III. REVIEW OF LEARNING METHODS FOR HIDDEN UNITS	58
The Connectionist Learning Problem	58
Direct Search	60
Assignment of Credit	62
Exact Gradient Methods	62
Approximate Gradient Methods	65
Minimal Change	72
Worth	77
Modification	80
Fraction of Gradient	81
Minimal Change	81
Generate New Units	82
Outline	85
Direct Search	85
Gradient Methods	86
Minimal Change	88
Worth	90
 IV. COMPARISON OF METHODS FOR LEARNING MISSING FEATURES	95
Direct-Search Algorithms	98
Unguided Random Search	98
Guided Random Search	102
Polytope Algorithm	105
Error Back-Propagation Algorithms	110
Rosenblatt	110
Rumelhart, Hinton, and Williams	112
Reinforcement Learning	117
Associative Search with Reinforcement Prediction	117
Associative Reward-Penalty	121

Local Reinforcement	123
Penalty Prediction	126
Summary	130
V. STRATEGY LEARNING WITH MULTILAYER CONNECTIONIST SYSTEMS	139
Strategy Learning Behavior of the Algorithms	140
Initial Search Strategy	140
Credit Assignment	141
Modification	144
Connectionist Algorithms for Strategy Learning	145
Output Functions	147
Learning Algorithms	150
Parameters	155
VI. LEARNING A SOLUTION TO A NUMERICAL CONTROL TASK	156
The Pole-Balancing Task	157
Control-Engineering Approach	159
Our Approach	160
Experiments	162
Simulation	162
Desired Functions	162
Interaction between Learning System and Cart-Pole Simulation	164
Results of One-Layer Experiments	166
Results of Two-Layer Experiments	171
Transfer to Similar Tasks	179
Conclusion	180
VII. LEARNING THE SOLUTION TO A PUZZLE	183
The Tower of Hanoi Puzzle	183
Experiments	187
Representation of States and Actions	188
Credit Assignment	189
Interaction between Learning System and Puzzle	190
Results of One-Layer Experiments	192
Results of Two-Layer Experiments	196
The Evaluation Function and New Features	198
The Action Function	203
Transfer of Learning	204
Conclusion	207

VIII. SUMMARY AND FUTURE WORK	210
Comparison of Methods for Learning in Hidden Units	210
Strategy Learning with Multilayer Connectionist Systems	213
Pole Balancing	214
Tower of Hanoi	216
Future Work	218
Applications	219
APPENDIX	
A. RESULTS FROM ONE-LAYER NETWORK EXPERIMENTS OF CHAPTER II	221
B. DERIVATION OF $P(a_j = 1)$ FOR TOWER OF HANOI EXPERIMENTS	225
Two Legal Actions	225
Three Legal Actions	227
C. WEIGHTS FOR SOLUTIONS OF STRATEGY LEARNING TASKS	229
Pole-Balancing Task	229
Tower of Hanoi Task	229
BIBLIOGRAPHY	234

LIST OF TABLES

1. Performance of Single Unit on Multiplexer Task	36
2. Performance of Single-Layer System on Input-Cluster Task	43
3. Performance of Single Layer System on Output-Vector Task	51
4. Unguided Random Search on the Multiplexer Task	101
5. Guided Random Search on the Multiplexer Task	104
6. Polytope Algorithm on the Multiplexer Task	109
7. Rosenblatt's Algorithm on the Multiplexer Task	112
8. Rumelhart, et al.'s Algorithm on the Multiplexer Task	115
9. Rumelhart, et al.'s Modified Algorithm on the Multiplexer Task	117
10. Associative Search with Reinforcement Prediction on the Multiplexer Task	121
11. A_{R-P} Algorithm on the Multiplexer Task	123
12. A_{R-P} with Local Reinforcement on the Multiplexer Task	125
13. A_{R-P} with Penalty Prediction on the Multiplexer Task	129
14. Performance Summary for Multiplexer Task	133
15. New Features Developed by Rumelhart, et al.'s Algorithm	135
16. New Features Developed by A_{R-P} with Penalty Prediction Algorithm	137
17. Sample Interaction of Learning System and Cart-Pole Environment	166
18. Results of One-Layer System	167
19. Results of Two-Layer System	172
20. Interaction of Learning System and Tower of Hanoi Puzzle	192
21. Results of One-Layer System	193
22. Results of Two-Layer System	197
23. Multiplexer Task—Parameter Search Results	221
24. Input-Cluster Task—Parameter Search Results	224
25. Expressions for $P(X > Y)$ by Cases	226

26. Expressions for $P(X > Y, X > Z)$ by Cases	228
--	-----

PREVIEW

LIST OF FIGURES

1. A Linear Threshold Unit	4
2. Discrimination of a Linear Threshold Unit	5
3. Model of a Two-Layer Learning System	7
4. Adding a New Term to a Symbolic Representation	23
5. Possible Concepts using Connectionist Representation	25
6. Adding a New Feature to the Connectionist Representation	26
7. Multiplexer Task—Original Representation	32
8. Multiplexer Task—Basis-Vector Representation	34
9. Multiplexer Task—Original Representation Plus New Features	35
10. Multiplexer Task—Learning Curves	37
11. Input-Cluster Task—Original Representation	40
12. Input-Cluster Task—Basis Vectors as Class Labels	41
13. Input-Cluster Task—Class Labels with Common Feature	42
14. Input-Cluster Task—Learning Curves	44
15. Output-Vector Task—Original Representation	49
16. Output-Vector Task—Hidden Units That Correlate Output Units	50
17. Learning Curves for the Output-Vector Task	52
18. Multi-Level Reasoning Network (from Fu and Buchanan [1985])	53
19. Naming Switchover Points	55
20. A Connectionist Network	59
21. The Logistic Distribution	63
22. Back-Propagation of Gradient Information	64
23. Two-Layer Network for Multiplexer Experiments	96
24. Learning Curves of Direct-Search Algorithms on the Multiplexer Task	102
25. Learning Curve for Error-Correction Algorithms on the Multiplexer Task	112

26. Learning Curve for Reinforcement Learning Algorithms on the Multiplexer Task	120
27. Learning Curves for All Algorithms on the Multiplexer Task	132
28. Two-Layer Networks for Strategy Learning	146
29. The Pole-Balancing Environment	157
30. Desired Functions for Pole-Balancing Solution	163
31. Balancing Time versus Trials for One-Layer System	168
32. Weights Learned by One-Layer Network	169
33. Functions Learned by One-Layer Networks	170
34. Balancing Time versus Trials for Two-Layer System	173
35. Weights Learned by Two-Layer Network	174
36. Functions Learned by Two-Layer Networks	176
37. New Features Learned by Two-Layer Networks	178
38. Initial and Goal States of the Tower of Hanoi Puzzle	184
39. State Transition Graph for Three-Disk Tower of Hanoi Puzzle	185
40. 3-Dimensional State Space and Desired Evaluations	186
41. Length of Solution Path versus Trials for One-Layer System	194
42. Weights Learned by One-Layer Network	195
43. Evaluation Function Learned by One-Layer Network	196
44. Length of Solution Path versus Trials for Two-Layer System	198
45. Weights Learned by Two-Layer Network	199
46. Evaluation Function Learned by Two-Layer Network	200
47. New Features Learned by Two-Layer Evaluation Network	201
48. State Transition Graph for Four-Disk Tower of Hanoi Puzzle	206
49. Output-Vector—Original Representation	222
50. Output-Vector—New Output Representation	223
51. Pole-Balancing—Weights Learned by One-Layer Network	230
52. Pole-Balancing—Weights Learned by Two-Layer Network	230
53. Tower of Hanoi—Weights Learned by One-Layer Networks	231
54. Tower of Hanoi—Weights Learned by Two-Layer Evaluation Network	232
55. Tower of Hanoi—Weight Values Learned by One-Layer Action Network with Two-Layer Evaluation Network	233

CHAPTER I

INTRODUCTION

Connectionist systems embody a framework for decision-making based on an active form of knowledge representation. They are composed of simple computational units interconnected by pathways that transmit numerically-valued signals rather than complicated symbolic messages. The connectionist framework stems both from psychological theories of how the brain represents associations among concepts and from the modeling of neural networks. Current methods and applications of connectionist systems constitute a paradigm at a level between symbolic and neural representations. Although connectionist and symbolic representations are not on the same level, they are subject to analogous problems concerning the development of a representation. *Representation development* is the process whereby modifications are made to a representation by the addition, removal, or alteration of the representation's components, whether these components are symbolic terms or numerically-valued features.

Interest in connectionist systems has grown in recent years for several reasons. The inherent parallelism of connectionist systems can result in fast decision-making. Also, the use of connectionist systems as models of cognitive processes has met with some success. For example, Rumelhart and McClelland's (1986) connectionist system models the verb-tense learning behavior of children, a behavior often presented as an argument for the symbolic representation of explicit rules. A third reason for the growing interest in connectionist systems is the recent progress towards a solution to the problem of learning in multilayer connectionist systems, overcoming an obstacle that has been a major criticism of the

connectionist paradigm supported by Minsky and Papert's (1969) analysis of the limitations of the perceptron (Rosenblatt, 1962).

The issues of learning in connectionist systems, however, are far from resolved. The work reported in this dissertation addresses three pressing learning issues:

1. relationships among various approaches to learning in multilayer connectionist systems,
2. the *structural* credit-assignment problem (defined below), and
3. learning in cases in which the desired output of the system is unknown.

The relationships among learning algorithms for multilayer connectionist systems are elucidated by reviewing them within a framework based on a categorization of methods for structurally assigning credit. Numerical learning algorithms not originally presented as connectionist learning algorithms are also discussed within this framework. Such a consistent review is much needed in a field where researchers come from diverse backgrounds, as is the case for connectionist learning research.

Some of the better-known learning algorithms for multilayer connectionist systems are studied by applying them to the multiplexer learning task (described in Chapters II and IV). In comparing the performances of different algorithms, consideration was given to details such as consistency in the training procedure, optimization of the algorithms' parameters, measures both for the performance level during learning and at the conclusion of a learning run, and statistical confidence intervals for all data. Such careful comparisons rarely appear in the literature but are necessary for drawing significant conclusions.

The third issue listed above is discussed in terms of reinforcement-learning methods, as described later in this chapter. To date, research with reinforcement-learning methods has focused on single-layer learning systems (Barto and Anandan, 1985; Barto, Sutton, and Anderson, 1983; Barto, Sutton, and Brouwer, 1981;

Barto and Sutton, 1981), though their potential use for learning in multilayer connectionist systems has been demonstrated in several small examples (Anderson, 1982; Barto, 1985; Barto, Anandan, and Anderson, 1986; Barto and Anderson, 1985; Barto, Anderson, and Sutton, 1982). In this thesis, reinforcement-learning methods are combined with a learning algorithm for multilayer systems to develop an example of a multilayer connectionist system for the learning of problem-solving strategies. This system is demonstrated on a pole-balancing control task and on a Tower of Hanoi puzzle.

Connectionist Systems

Connectionist systems generally consist of a collection of computational units, sometimes described as neuron-like in their input-output behavior. Each unit receives a number of input signals, or *input components*, whose numerical values constitute the unit's *input vector*, and the unit applies an *output function* to its input to generate output values. *Networks* of units are constructed by connecting the output of some units to the input of other, or the same, units. A network is said to interact with an *environment* by receiving a vector of numerical values from the environment and producing an output vector that acts upon the environment. Thus a unit's input components can originate either from the network's environment or from the output of another unit; a unit's output can be passed on to another unit or it can become a component of the network's output.

A unit's output function is parameterized by a vector of numerical weights, one weight for every input component. For a given network structure, it is the values of these weights that determines the input-output behavior of the network. A *learning algorithm*¹ for a connectionist system is a method for updating the

¹ "Algorithm" is not used here in the strictest sense of Knuth's (1973) definition: finiteness is not assumed. Knuth suggests that such procedures be called "computational methods", but for brevity we will use "algorithm." Also, by using the label "learning algorithm" we do not imply the existence of a proof of convergence to a desirable outcome.

values of the system's weights based on the performance of the system. For reviews of current connectionist research see Feldman and Ballard (1982), Hinton and Anderson (1981), McClelland and Rumelhart (1986), and Rumelhart and McClelland (1986).

The output functions performed by the units are usually one of a small set of functions. The most common function is the linear threshold function used by the pioneers of adaptive networks (e.g., Farley and Clark, 1954; McCulloch and Pitts, 1943; Rosenblatt, 1962; Widrow, 1962). A unit implementing this function is sketched in Figure 1, where x_1, x_2, \dots, x_n are the components of the unit's n -

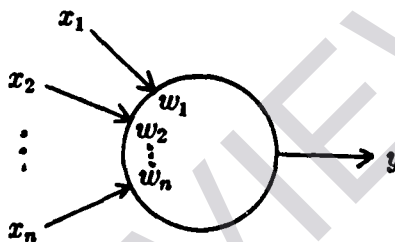


Figure 1: A Linear Threshold Unit

dimensional input vector, w_1, w_2, \dots, w_n are the unit's weights, and y , the unit's output, is defined as

$$y = \begin{cases} 1, & \text{if } \sum_{i=1}^n x_i w_i > 0; \\ 0, & \text{otherwise.} \end{cases}$$

The threshold of this unit is zero, but it can be considered to be parameterized if one of the input components is constant (so that the value of the corresponding weight becomes the negative of the threshold²). Such a unit can discriminate (i.e., produce different output values for) sets of input patterns that are linearly-separable (can be separated by an $n - 1$ dimensional surface). For example, a unit receiving two input components discriminates input vectors by a line, as shown

²Using a constant input is a standard technique for implementing a variable threshold. See Nilsson (1965) for further details.