1992

A Critical Examination Of Connectionist Cognitive Architectures

Marin S. Marinov

A CRITICAL EXAMINATION OF CONNECTIONIST COGNITIVE ARCHITECTURES

by

Marin S. Marinov

Department of Philosophy

Submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy

Faculty of Graduate Studies
The University of Western Ontario
London, Ontario
July, 1992

ISBN 0-315-75375-7

Abstract

The dissertation represents a critical evaluation of the major connectionist theories of human cognitive architecture. The central connectionist thesis that artificial neural networks (ANNs) can serve as finitary models of human cognizers is examined and rejected. Connectionist theories, in contrast to the classical symbol-processing theories of cognitive architecture, cannot explain the productivity and systematicity of mental states. The reason for this is that ANN-based cognitive architectures cannot maintain representational states with compositional structure. Chapter One analyzes the implementational connectionism's solution to the problem of compositionality. It is shown that neither the theory of weak nor of strong compositionality can solve this problem.

Chapter Two criticises the attempt to establish connectionism as an alternative theory of human cognitive architecture through the introduction of the symbolic/ subsymbolic distinction. The reasons for the introduction of this distinction are examined and found to be unconvincing. Several experimental comparisons between the TDIDT class of symbolic learning systems and the class of artificial neural networks using the error backpropagation algorithm are discussed. It is argued that the differences in the performance of these two classes of learning systems are insignificant and are not systematic. Such evidence

contradicts the view that ANNs define a new kind of "subsymbolic" computation.

Supporters of eliminative connectionism have argued for a pattern association and pattern recognition-based explanation of cognitive processes. They deny that explicit rules and symbolic representations play any role in cognition. Their argument is based to a large extent on Rumelhart and McClelland's and MacWhinney and Leinbach's connectionist models of learning of the past tenses of English verbs. Chapter Three presents an analysis of an experimental comparison between these models and the Symbolic Pattern Associator (SPA) -- a learning system based on the classical architecture. It is shown that the SPA outperforms the connectionist models; moreover, the SPA can represent the acquired knowledge in the form of explicit rules. The analysis of this comparison leads to the conclusion that symbol-processing models have a far better chance of explaining complex cognitive phenomena in terms of rules and symbolic representations than eliminative connectionism.

ACKNOWLEDGEMENTS

I have benefitted greatly from discussions with my thesis supervisor William Demopoulos. Without his intellectual and moral support this dissertation would have been impossible.

I owe a great deal to Zenon Pylyshyn, Ausonio Marras, Charles Ling, Michael Devitt, Keith Humphrey, and Bruce Freed who had read earlier drafts of my thesis and made very helpful critical comments.

I am deeply indebted to my teachers John L. Bell, Robert Butts, Michael Dawes, William Demopoulos, Lorne Falkenstein, Bruce Freed, William Harper, Keith Humphrey, Edward Stabler, Jr., and John Thorp.

I benefitted a great deal from the work on the SPA together with Charles Ling and Steve Cherwenka.

My Department provided the right intellectual atmosphere and supported me generously throughout my graduate years.

Last but not least, I want to acknowledge the very strong moral support of my wife Miglena Nikolchina. Her help was indispensable throughout the four long years at Western.

TABLE OF CONTENTS Page

CERTIFICATE OF EXAMINATION ii

ABSTRACT iii

ACKNOWLEDGEMENTS v

TABLE OF CONTENTS vi

LIST OF FIGURES ix

LIST OF TABLES x

MOTTO xi

INTRODUCTION 1

CHAPTER ONE - THE CASE AGAINST CONNECTIONISM:
PRODUCTIVITY AND SYSTEMATICITY 11

1. Introduction 11
2. Productivity 12
   - Finitary Models of Human Cognizers 12
   - Artificial Neural Networks and
   Turing Machines 14
3. Systematicity 22
   - Cognitive Architecture and the Structure
   of Representational States 22
   - Weak Compositionality 34
   - Strong Compositionality 43
4. Conclusion 54

CHAPTER TWO - ON THE SYMBOLIC/ SUBSYMBOLIC
DISTINCTION 55

1. Introduction 55
2. The Reasons for the Introduction of the
Symbolic/ Subsymbolic Distinction 56
3. The Brain-Likeness of Error Backpropagation
in Artificial Neural Networks 60
4. The Problem of Brittleness 63
5. The ID3 Symbolic Learning System 64
- How to Avoid Brittleness 72
- Commonsense 76
- On the Notion of Subsymbolic Computation 84
6. Conclusion 92

CHAPTER THREE - ANSWERING THE ELIMINATIVIST CHALLENGE:
THE IMPORTANCE OF EXPLICIT REPRESENTATIONS 94

1. Introduction 94
2. Eliminativism 95
3. Eliminative Connectionism 98
4. An Examination of the Eliminative Connectionist
Conception of Language Learning and
Language Processing 109
5. Rumelhart and McClelland's Model of Learning
the Past Tenses of English Verbs 114

- Criticism of Rumelhart and McClelland's Model 120
- Performance Results 124
6. MacWhinney and Leinbach's Model of Learning the Past Tenses of English Verbs 128
- Results 133
- Criticism of MacWhinney and Leinbach's Model 137
7. Is There a Better Symbolic Model? 146
8. The Symbolic Pattern Associator 147
- The Requirements for the Model 148
- The Architecture of the Symbolic Pattern Associator 151
- Experimental Set-Up 158
- Performance of the Symbolic Pattern Associator on the Past Tense Learning Task 166
- Explicit Representations and Higher-Level Processing 177
9. Conclusion 188

ENDNOTES 190
APPENDIX I: GLOSSARY OF ACRONYMS 193
BIBLIOGRAPHY 195
VITA 205

LIST OF FIGURES

Figure 1.1. Representation of "coffee" 36
Figure 1.2. Representation of "cup without coffee" 38
Figure 1.3. Representation of "cup with coffee" 40
Figure 1.4. The tensor product representation for filler/role bindings 46
Figure 2.1. Decision tree for predicting weather conditions 69
Figure 2.2. Decision tree for predicting party affiliations 79
Figure 2.3. Pruned decision tree for predicting party affiliations 82
Figure 3.1. Explanatory understanding as the activation of a prototype 104
Figure 3.2. Rumelhart and McClelland's network 116
Figure 3.3. MacWhinney and Leinbach's network 134
Figure 3.4. SPA: A joint decision method for associating arbitrary patterns 155
Figure 3.5. Typical phonetic decision tree created during the learning process 164

LIST OF TABLES

Table 2.1. Golf Data Set 67

Table 2.2. ID3 Performance on Several Real-World Tasks 74

Table 2.3. Votes on Proposed Legislation 78

Table 2.4. Experimental Comparison of ID3 and an ANN Using BP 87

Table 2.5. Experimental Comparison of CART and an ANN using BP 90

Table 3.1. Propositions and Truth-Values 107

Table 3.2. Binary Patterns to Be Associated by the SPA 153

Table 3.3. Pairs of Phonetic Patterns for Training 162

Table 3.4. Results of the Performance of the SPA on the Learning of the Past Tenses of English Verbs 167

Table 3.5. Types of Responses 169

Table 3.6. A comparison between SPA and the Connectionist Models in Learning of the Past Tenses of English Verbs 176

Table 3.7. Inductive Steps in the Learning of the Past Tenses of English Verbs 187

MOTTO

Connectionism has dramatically shifted the mainstream of opinion in cognitive science, but only because the existing implementations actually perform nontrivial tasks in ways unanticipated by recent opinion. If those of the "classical" school want to resume their hegemony, they will need more than a persuasive ideology -- they have had that all along -- they will need some positive results of actual modelling as striking as those the connectionists have used to attract our attention.

Daniel Dennett

INTRODUCTION

The notion of cognitive or functional architecture is one of the central notions in cognitive science. It has assumed a particular importance in the debate between the two major schools of thought in contemporary cognitive science -- classicism and connectionism. Although there is not complete consensus on its definition, the cognitive or functional architecture of a computational device can be roughly defined as the fixed resources that allow the device to operate on structures for which it is theoretically useful to assume some semantic interpretation. For example, if a computational device can take as input the codes for a & (a --> b); (aVc) & ~a; a & (a -->b) & (c --> d) & (~c --> ~b) and output the codes for b, c, and d respectively, and if it can process a large number of such codes, we have good reason to assume that it has a capacity to manipulate the codes of propositional logic formulas which can receive a semantic interpretation. Given sufficient time and effort, we can find out exactly which operations of this computational device underlie its capacity to manipulate propositional representations. These operations can be factored into operations that are specific for this task and into the set of the fixed resources of the device that constitute its cognitive or functional architecture.

As a first approximation, the fixed resources of a computational device that constitute its cognitive

architecture include the set of the basic operations that enable the device to manipulate representations. Minimally this set should include operations for sorting and retrieving representations, for comparing them, and for treating them differentially as a function of how they are stored. The fixed resources of the device determine such processing limitations, as limited memory and the relative complexity of a given type of operation vis-a-vis the set of basic operations. The cognitive architecture includes also the control structure which matches appropriate operations for different tasks (cf. Pylyshyn, 1984, pp. 30-31; See also: Newell, 1980; Newell, Rosenbloom, and Laird, 1989). In short, the cognitive architecture can be seen as a kind of user manual for a given computational device in direct analogy with the user manuals provided for programming languages. And if the mind/ brain can be viewed as a kind of computational device, the theory of human cognitive architecture should be able to give us the basic operations, limitations, and control structures of the human mind. In this sense the theory of human cognitive architecture is relevant to all theories that attempt to uncover the fixed resources of the mind and to outline its limits. The theory of cognitive architecture is therefore the cornerstone of any computational theory of the mind.

Until recently, the theory of human cognitive architecture was not a hotly contested issue in cognitive science. Most cognitive scientists shared the common view that

since the mind is best understood as a kind of computational device, its functional architecture must be similar to the functional architecture of other, better understood, computational devices such as the von Neumann computer. Following Chomsky and Miller (1963), Putnam (1960), and others it was thought that the functional architecture of the Turing machine (TM) can serve as the best model of human cognitive architecture.

Another founding principle in cognitive science has been that computational devices based on the Turing/ von Neumann cognitive architecture are symbol manipulating mechanisms. Because human minds and von Neumann computers can manipulate codes with semantic interpretations it has been assumed that these mechanisms can best be described as physical symbol systems (Newell, 1980; Pylyshyn, 1984). The consensus reached among the majority of cognitive scientists on both of these issues was seen as the cornerstone of the computational theory of the mind and has come to be known as the 'classical' or 'symbol-processing' approach to cognition.

This peaceful state of affairs was disrupted with the advent of artificial[1] neural networks (ANNs). ANNs in the form of Rosenblatt's perceptrons (Rosenblatt, 1962) have been around almost since the beginning of the computer revolution. But it was not until the early 1980s that ANNs assumed a central stage in the development of cognitive science. This period coincided with the development of multilayered

perceptrons and with significant improvements in their learning algorithms. The multilayer ANNs were shown to have some unexpected computational properties in comparison with their two-layer predecessors. Thus, it was shown that multilayer ANNs can solve the XOR problem, a problem which is unsolvable for the two-layer perceptron[2]. This held out the promise of a wide range of future applications using multilayer ANNs.

Interest in the engineering applications of ANNs gave rise to a new foundational debate and a new movement in cognitive science and in philosophy of mind -- connectionism. What is at stake in the debate between the connectionists and the supporters of the classical symbol processing approach to cognition is nothing more nor less than the 'right' computational model of the mind/ brain or the nature of the 'right' human cognitive architecture. Connectionists claim that the ANN, and not the TM, should be seen as providing the best model for the mind/ brain and that human cognitive architecture is entirely ANN-based.

The debate between classicists and connectionists has centered on several important issues. One issue is the choice of criteria for selecting either the TM or the ANN as the basis for human cognitive architecture. In particular, the answers to three questions are of major significance:

1. What are the adequacy conditions that finitary models of human cognizers must meet?

2. What are the computational differences between TMs and ANNs?

3. How well can the rival theories of cognitive architecture account for the compositional character of mental states?

In Chapter One I show that the answer given by the connectionist theorists to each of these questions is less than satisfactory. Connectionist research has largely been confined to the study of the properties of particular connectionist models designed to simulate some very restricted cognitive tasks. These simulations have often been used to justify the claim that the cognitive architecture of the mind as a whole is connectionist. It is doubtful, however, that the performance of these connectionist models can be used to justify such a theoretical conclusion. The root of the problem is that ANNs cannot serve as adequate finitary models of human cognizers. Although one can point to some connectionist models that have performed relatively well, there are certain pervasive cognitive phenomena, like productivity and systematicity, that cannot be explained by connectionist theories of cognitive architecture.

So far, there has been no attempt to explain how ANN-based cognitive architectures can account for the phenomenon of productivity -- the fact that a native speaker of a language has the ability to comprehend and produce on

appropriate occasions an immense number of previously unencountered sentences. In Chapter One I criticize the widespread belief that since ANNs can simulate TMs, they can also serve as models of human cognizers. There are several formal results which demonstrate that any ANN with a finite number of 'neurons' which are capable of being only in a finite number of states cannot be computationally universal and therefore cannot simulate TMs. The only way ANNs can be shown to be TM-equivalent is if either one of these finitary restrictions is negated, i.e. if an ANN has either an infinite number of 'neurons', or 'neurons' that can be in infinite number of states. But ANNs with an infinite number of 'neurons', or equivalently, with an infinite number of states cannot serve as finitary models of human cognizers.

Another pervasive cognitive phenomenon is the phenomenon of systematicity -- the fact that human thoughts and expressions are systematically related to one another. The problem at the heart of this issue is how ANN-based cognitive architectures can support representational states with a compositional structure. So far the only attempts to explain systematicity from a connectionist perspective have been Smolensky's theories of weak compositionality and of strong compositionality (Smolensky, 1987, 1991). However, Smolensky's theory of weak compositionality cannot explain how ANNs can be in representational states that are truly compositional. The operation of vector addition and vector subtraction that

underlies the theory of weak compositionality cannot guarantee the compositionality of connectionist representations. The same is true of the theory of strong compositionality, which relies on the operation of tensor product. The failure of connectionist theories of cognitive architecture to solve the problems of productivity and systematicity indicates that on purely theoretical grounds there is no reason to abandon the view that the Turing/ von Neumann cognitive architecture provides the best explanation of both of these pervasive cognitive phenomena.

Another very important issue that is shaping the debate between classicists and connectionists concerns the existence of "subsymbolic" computation, the computation that is supposed to be the hallmark of "connectionist computation". If there is such a new mode of computation -- one that is different in kind from the classical Turing machine-defined computation -- then we would have prima facie evidence that the classical theory of cognitive architecture might be incomplete. However, the arguments given in support of the existence of a new kind of subsymbolic computation are defective. In Chapter Two, I present a detailed critique of these arguments and I demonstrate that computational models based entirely on the classical symbolic architecture are quite capable of performing "subsymbolic" computations, thus showing the spuriousness of Smolensky's introduction of this concept (Smolensky, 1987, 1988). In particular, I analyze several

experimental comparisons between the class of ANNs using the error backpropagation learning algorithm and the Top-Down Induction of Decision Trees (TDIDT) class of symbolic learning systems. This analysis demonstrates that there exists a class of symbolic systems based on the Turing/ von Neumann cognitive architecture that, on the same type of learning task, can perform as well as, if not better than, the class of ANNs. Therefore, the claim that there exists a special "subsymbolic" form of computation -- allegedly characteristic only of ANNs -- is discredited.

It may be misleading to speak of connectionism as presenting a single theory of human cognitive architecture. There exists a deep disagreement in the connectionist camp on the issue of what constitutes an ANN-based cognitive architecture and to what an extent ANN-based architectures can or should implement symbolic-processing architectures. The majority of connectionists are divided into two schools of thought -- implementational connectionism and eliminative connectionism.

The position of the implementational connectionism is that although ANN-based and TM-based cognitive architectures are strictly speaking incompatible, ANN-based architectures can implement TM-based symbolic architectures. The position of the eliminative connectionism is that instead of trying to implement symbol-processing architectures, connectionist theory should demonstrate that there is no need to postulate

the existence of symbol-processing operations in the human cognitive architecture.

There are a number of difficulties associated with both of these views. If ANNs could implement TMs, as the first school of thought insists, then ANN-based architectures are either irrelevant for psychology as 'mere' implementations of the classical symbolic architecture, or inadequate as approximate implementations of the classical architecture. Even though the supporters of implementational connectionism claim that they can somehow steer between the horns of this irrelevance/ inadequacy dilemma, the majority of 'orthodox' connectionists regards the attempt to implement in neural hardware symbol-processing structures as a blind alley. They believe that ANN-based cognitive architectures leave no place for symbolic structures like propositions, production rules, parse trees, part-whole hierarchies, etc. The supporters of eliminative connectionism deny that rules and symbolic representations play any role in cognition and they foresee their gradual elimination from cognitive science. Eliminative connectionism implies a total rejection of the classical cognitive architecture and thus a rejection of the hope that folk psychology can be successfully reduced by the mature cognitive science; eliminative connectionism implies the elimination, rather than the reduction, of the central terms of folk psychology.

The arguments in favour of eliminative connectionism have been almost exclusively based on concrete demonstrations of connectionist models that try to solve some important cognitive tasks. Although the majority of these models have been offered simply as illustrative models, there have been some connectionist models that try to solve serious nontrivial cognitive tasks. Two of the most important are Rumelhart and McClelland's (1986) and MacWhinney and Leinbach's (1991) models of learning the past tenses of English verbs. In Chapter Three I examine critically the most important claims of eliminative connectionism and I find them to be unfounded. In particular, I demonstrate that there are symbolic learning models that can outperform the existing connectionist models on the task of learning the past tenses of English verbs; moreover, these models hold out the promise of actually outperforming ANN-based models on nontrivial cognitive tasks. In marked contrast to the connectionist models the 'Symbolic Pattern Associator' can express the acquired knowledge in explicit form and can point the way to integrating this knowledge systematically. No corresponding capabilities are present in the eliminativist models. The existence of such superior models based on the Turing/ von Neumann architecture demonstrates the falsity of the argument in favour of eliminative connectionism.

CHAPTER ONE

The Case Against Connectionism: Productivity and Systematicity

Introduction

One of the main aims of cognitive science is to provide a finitary model of human cognizers with which to explain cognitive phenomena. The debate in cognitive science between connectionists and supporters of the so called classical approach can be put in terms of a simple dilemma:

(1) Classicists: The finitary model of human cognizers is the Turing Machine (TM).

(2) Connectionists: The finitary model of human cognizers is the artificial neural network (ANN).

In order to gain an understanding of the nature of the debate and to be able to judge which position is the correct one we have to explore three questions:

- What are the adequacy conditions that finitary models of human cognizers must meet?
- What are the differences between TMs and ANNs?
- How do the cognitive architectures that the different finitary models specify explain certain essential properties of mental states?

Productivity

Finitary Models of Human Cognizers

Any model of human cognizers must meet some minimal adequacy conditions. Such a model must be powerful enough to explain pervasive cognitive phenomena such as productivity and systematicity and at the same time it must be finitely specifiable. Let us first examine productivity.

It can be argued that contemporary cognitive science began with the realization of the productivity or unboundedness of human cognitive capacities. For example, in 1963 N.Chomsky and G.Miller wrote:

> The fundamental fact that must be faced by any investigation of language and linguistic behaviour is the following: a native speaker of a language has the ability to comprehend an immense number of sentences that he has never previously heard and to produce, on appropriate occasions, novel utterances that are similarly understandable to other native speakers (Chomsky and Miller, 1963, p.271).

The words immense number of sentences should not be taken lightly: G.Miller has estimated that the number of well-formed 20-word English sentences is on the order of magnitude of the number of seconds in the history of the universe (cf. Fodor and Pylyshyn, 1988, p.24).

This fundamental fact has led N.Chomsky and others to conclude that our capacity to produce and understand language

is potentially unbounded or productive. No one can utter more than a finite number of sentences due to finite memory, limited lifespan, and the limitations of the mechanisms underlying speech. But in order to explain language use we have to introduce the idealization that human linguistic competence is productive. So, an appropriate model of human language users should reflect both the potential unboundedness of linguistic competence as well as the finiteness of the resources that interact with competence to produce actual linguistic performance. A model of human language users that reflects these constraints must be able to isolate the finite productive mechanism responsible for producing and recognizing a potentially infinite number of well-formed expressions. As Chomsky and Miller insisted, an automaton that compiles a simple list of all the grammatical sentences it hears simply cannot be considered an adequate model. Such a finite state automaton can learn the potentially infinite set of grammatical sentences only by adding an infinite number of new states. This however violates the finiteness restriction on the model of human language users because the capacity to produce and understand a potentially infinite number of sentences must be finitely represented.

In contrast with a model based on a finite state automaton, a model of human language users based on the Turing machine clearly meets this finite representability condition. A TM can serve as a finitary model of human language users

because its productive capacities are potentially infinite while being finitely representable. A TM running a finitely specifiable program can nonetheless operate on an infinite number of sentences by virtue of having a potentially infinite tape.

The argument that the productivity of linguistic capacities extends to human thinking and cognition in general has been made by Fodor (1975). According to his language of thought hypothesis, the ability to produce and understand an unbounded number of sentences is clearly correlated with the ability to think a potentially unbounded number of thoughts. Since language and thought are the paradigmatic cognitive capacities, we hypothesize that human cognition is <u>in general</u> productive. But if human cognitive capacities are productive we have at least one very good reason for maintaining that the TM is an adequate finitary model of human cognizers. Do we have comparably good reasons for accepting connectionism? In order to answer this question we have to look more carefully into the relation between TMs and ANNs.

<u>Artificial Neural Networks and Turing Machines</u>

One often reads claims to the effect that ANNs can simulate TMs or are computationally equivalent to TMs, or that ANNs can compute the class of Turing-computable functions. For example Rumelhart and McClelland state that

...one can make an arbitrary computational machine out

> of linear threshold units, including, for example, a machine that can carry out all the operations necessary for implementing a Turing machine. (Rumelhart and McClelland, 1986, p.119.)

In a similar vein Smolensky tells us that "It is well known that von Neumann machines and connectionist networks can simulate each other" (Smolensky, 1988, p.7).

Such claims are clearly aimed to serve as a premise for the following argument

> 1. TMs have been used as models of human cognizers
> 2. ANNs can simulate TMs.
> ---
> Therefore, ANNs are at least as good as models of human cognizers as are TMs.

Indeed, many connectionists (including Smolensky (1988), Rumelhart and McClelland (1986)) argue for the stronger claim that ANNs are actually <u>better</u> models than TMs.

Let us begin by examining in greater detail what reasons there are for supposing that ANNs can simulate TMs. Optimism on this issue has been fuelled by a growing battery of formal results that aim to prove that artificial neural networks are computationally equivalent to Turing machines. There exist several proofs of this claim using different types of ANN. But all such proofs have to meet one major obstacle: The computational power of any ANN with <u>a finite number of units (neurons)</u> which are capable of being in only <u>a finite number</u>

of states does not exceed the power of a finite state automaton. But then it follows by a well established result of automata theory that ANNs with only a finite number of neurons which are capable of only a finite number of states cannot be computationally universal, and therefore cannot be TM-equivalent, i.e. there exist functions that are Turing computable but that are not computable by any (finite) ANN (cf. Hartley and Szu, 1987). Thus, all proofs of the Turing machine equivalence of ANNs have to relax at least one of these two restrictions. And in fact we do find that all proofs of TM-equivalence assume either an infinite number of neurons or neurons with countably or uncountably infinitely many states, even though the actual details of the proofs may differ widely.

For example, Goles and Martinez (1990) assume infinitely many neurons. They prove the computational universality of ANNs via cellular automata, assuming an infinite number of cells (neurons) in the cell space. Goles and Martinez treat ANNs as cellular automata whose graphs have a weighted structure. They prove that ANNs can simulate standard cellular automata. Then, using the well known proofs of the equivalence of certain cellular automata having infinitely many cells with Turing machines, they are able to prove the equivalence of ANNs with an infinite number of neurons and TMs (Goles and Martinez, 1990, pp. 20-26). A similar proof, relying on the postulation of an infinite number of neurons, is given by

Garzon and Franklin (1989).

A different approach is taken by Siegelman and Sontag (1991). They prove the equivalence of TMs to ANNs which have finitely many neurons each of which is capable of being in infinitely many states. Siegelman and Sontag prove that it is possible to construct a network that can simulate a push-down automaton with two binary stacks. Since it is well known that a TM can be simulated by a two stack push-down automaton (Hopcroft and Ullman, 1979, pp.172-73), it is possible to construct a network where neurons with infinitely many states can represent stacks capable of encoding unbounded information (Siegelman and Sontag, 1991, p. 2). Other proofs relying on the postulation of neurons with infinitely many states are given by Giles et al. (1990), and Sun et al. (1991). Thus, none of these results is really very surprising since allowing a finite state automaton to become an infinite state automaton makes it trivially computationally universal.

The question to be asked is how relevant are these results for the debate between classicists and connectionists. Two very obvious limitations of these mathematical results have to do with practical realizability:

(i) There can be no ANN built out of realistic components that has an infinite number of neurons, and

(ii) There can be no ANN built out of realistic components whose neurons are capable of being in an infinite number of states.

These practical limitations make all the difference in the world for the computational universality of actual ANNs. They show that the computational power of any realistic ANNs cannot exceed the power of finite state automata. So, actual ANNs cannot be TM-equivalent. Connectionists, however, are adamant that such finitary restrictions do not in any way undermine their program. They note that the same finite hardware restrictions apply to realistic computers and human beings alike. As Rumelhart and McClelland put it "... one limitation is that real biological systems cannot be Turing machines because they have finite hardware" (Rumelhart and McClelland, 1986, p.119).

Now, the reason why we say that actual finite von Neumann computers are effectively Turing machines is that we can always add more registers to their memory -- effectively, this means that we can expand the tape of the Turing machine. Many connectionists see a similarity between the finiteness and potential expandability of von Neumann computers and the finiteness and potential expandability of real- component ANNs. The reasoning here is patently analogical, as Siegelman and Sontag put it "...this potential infinity [unbounded number of neurons or unbounded number of neuron states] is analogous to the potentially infinite tape in a Turing machine" (Siegelman and Sontag, 1991, p. 2).

It is important to understand this analogy. From a purely mathematical point of view the two systems are analogous

because they share a common property -- they can both be in an infinite number of states and hence they can both be computationally universal. In other words, the potential expandability of ANNs -- the possibility of creating realistic networks with larger and larger numbers of neurons or neuron states -- is thought to be analogous to the potential expandability of the memory of von Neumann machines, and thus, to analogous potential expandability of the tape of a TM. However, from the point of view of cognitive science there is a deep disanalogy between them: infinite or potentially unbounded ANNs cannot serve as finitary models of human cognitive capacities because they are not finitely representable, whereas TMs can so serve because they are finitely representable.

TMs are finitely representable because a clear distinction can be drawn between the finite control unit of a Turing machine which has a finite description and the memory tape. The finite control unit of a TM specifies the transition regularities and the states which the machine can be in, but the possible extension of the memory tape does not in any way change its description. This is clearly not so in the case of ANNs. Every expansion in the memory of an ANN is bound to affect the structure of the network by adding new neurons and weights, thus changing its description. Thus, an infinitely expandable ANN is not finitely representable. Many researchers have simply assumed that if ANNs can be proven to be

computationally universal this is enough to justify their adoption as an adequate cognitive model. What they have missed is the patent inability of infinite-state ANNs to serve as finitary models of human cognizers.

The main reasons for the failure of the TM/ ANN analogy is that it is not the computational universality of ANNs that matters but how well ANNs or TMs can serve as finitary models of human cognitive capacities. This crucial difference can be missed if one looks only at the universality results. As Pylyshyn puts it

> ...it isn't the unboundedness per se that is important for our purpose, it is the form of organization the [unboundedness] condition imparts. After all, one could extend without limit the number of states of a finite state automaton; yet, without some finitary characterization of state-transition regularities, we would be right where we are in the case of an infinite-axiom system: Such a characterization would not allow us to understand (or, as Frege puts it, "survey") the function such a machine is computing....A device possessing an infinite number of (nonarticulated) states cannot be given an effective semantic interpretation because the mapping from states to a semantic model cannot be specified in a finitary manner. (Pylyshyn, 1984, p.72).

Pylyshyn's point is that TMs allow for a clear separation

between the finitely specifiable resources of the machine (the finite control unit and its transition table), the finitely specifiable program that it is running, and the possibly infinite set of data structures the machine can process. This allows us to vary the programs and the possible data structures a given Turing machine can process almost without limit. But the distinction between machine, program, and data structures cannot be drawn in the case of ANNs. Changes in data structures are bound to affect the constitution of any connectionist network:

> In a system such as a Turing machine, where the length of the tape is not fixed in advance, changes in the amount of available memory can be affected without changing the computational structure of the machine; viz., by making more tape available. By contrast, in a finite state automaton or a Connectionist machine, adding to the memory (e.g., by adding units to the network) alters the connectivity relations among nodes and thus does affect the machine's computational structure. Connectionist cognitive architectures, cannot by their very nature, support an expandable memory, so they cannot support productive cognitive capacities. (Fodor and Pylyshyn, 1988, pp.34-35.)

To summarize: If human cognitive capacities are productive, connectionist networks cannot offer a finitary

model of these capacities. So we have no option but to reject the view that ANNs are satisfactory finitary models of human cognizers.

Systematicity

Cognitive Architecture and the Structure of Representational States

We saw that if connectionists accept that human cognitive capacities are productive, then they must reject the view that the artificial neural network (ANN) can serve as a finitary model of human cognizers. But since, in order to establish the productivity or unboundedness of linguistic competence, we have to rely on a crucial idealization, connectionists are prepared to deny productivity. If cognitive competencies are considered strictly finite, despite all the explanatory difficulties this limitation would involve, the connectionist view can still be an alternative to the classical model. But there is another argument against taking the ANN as an adequate finitary model of human cognizers. And this argument does not require the idealization to unbounded cognitive capacities.

The TM model of cognitive systems implies a particular theory of the human cognitive architecture. The adoption of a particular cognitive architecture, in turn, imposes constraints on the type of representational states the system can be in. So, a particular cognitive model can be tested with

respect to the type of representational states that it can or cannot support. If it turns out that a given architecture does not support the kind of representations necessary to explain certain pervasive cognitive phenomena, then this architecture should be rejected as should the cognitive model on which it is based.

The cognitive or functional architecture of a computational device comprises the fixed resources that allow it to operate on structures that can be semantically interpreted. As a first approximation, the theory of these fixed resources include a specification of the set of basic operations that enable this device to manipulate representations. Minimally this set should include operations for sorting and retrieving representations, comparing them, treating them differently as a function of how they are stored, etc. The theory should define the relevant processing limitations, such as limited memory and complexity limitations; and finally it should specify the control structure which selects the rules to apply for different tasks (cf. Pylyshyn, 1984, pp. 30-31; See also: Newell, 1980; Newell, Rosenbloom, and Laird, 1989). In short, the cognitive architecture can be seen as a kind of user manual for a given computational device in direct analogy with the user manuals provided for programming languages. Cognitive models based on the TM and on the ANN will specify two sets of very different basic operations and limitations, and the opposition between

classicists and connectionists can be reformulated with respect to the different cognitive architectures that they envisage. For classicists the human cognitive architecture is the Turing/ von Neumann cognitive architecture e.g. the basic operations and limitations provided by the Turing machine, while for connectionists it is the functional architecture of the artificial neural network whose basic operations and limitations are seen as incompatible with those of the TM.

Since the basic operations and limitations of a cognitive architecture constrain the semantically interpretable states or representations of the system, the presence or absence of certain basic operations or limitations in a given functional architecture will make it possible (or impossible) to process certain types of representation. For example, the absence of the basic operation "retrieve from memory" will make it impossible to process representations not currently available to the system. In this way we can speak of the functional architecture as determining the range of possible representational states of a cognitive system. So, if the classical and the connectionist cognitive architectures are incompatible they will specify very different ranges of possible representational states.

In their influential article 'Connectionism and cognitive architecture' Fodor and Pylyshyn (1988) argue that the set of basic operations and functions provided by connectionist architectures is severely limited. These limitations render

connectionist architectures unable to support or process the type of representational state that is necessary to explain human behaviour.

Whether one is willing to accept productivity arguments or not, Fodor and Pylyshyn (1988) believe that it is fairly non-controversial that human cognitive capacities (finite or infinite) are systematic. And the best explanation of systematicity, they argue, is the existence of mental representations with compositional (combinatorial) structure. To see why this is so, let's look again at human language comprehension and human reasoning.

It is clear that our ability to produce and understand certain sentences is systematically related to our ability to produce and understand certain other sentences. The systematicity of human linguistic abilities can best be brought out if we consider what would be the case if these abilities were not systematic. If humans were able to learn to speak a language just by memorizing an enormous phrase book containing entirely unrelated phrases, the human capacity to understand language would in a genuine sense be non-systematic. For example, in order to understand such phrases as "raining cats and dogs", or "kicked the bucket", it is not necessary to know the meanings of their constituents; such phrases can be memorized as units since their meaning is not a function of their parts. If this were the case with all sentences of a natural language, then the learning of a

natural language would be an endless accumulation of unrelated idioms.

Of course, the phrase book example shows that this is not what is going on when humans learn to speak a language. As Fodor and Pylyshyn (1988, p.37) put it, "you can learn any part of a phrase book without learning the rest". If the phrase book story were even remotely plausible we could observe speakers of a language who are able to understand the sentence

The cat is on the mat.

and at the same time are unable to understand the sentence

The mat is on the cat.

But this is clearly not the case. Any mature native speaker of English who knows how to say

The girl loves John.

necessarily knows how to say

John loves the girl.

What makes the difference between a native speaker of English and a memorizer of a phrase book is that the native speaker has a knowledge of English syntax and semantics that the memorizer of the phrase book simply lacks. And given human memory limitations, we can figure out who is the memorizer fairly quickly. So if our linguistic capacities are systematic in this sense then knowledge of a language entails the ability to be in <u>systematically related mental states</u>. But being in systematically related mental states means that such mental

states have constituent parts that are systematically related to one another, i.e. some mental states are functions of others. So mental representations must have combinatorial structure. According to Fodor and Pylyshyn, connectionist mental states do not have combinatorial structure and therefore, connectionism cannot account for systematicity.

The argument from the systematicity of language understanding can be applied to thinking and reasoning as well. The ability to think the thought that aRb is systematically related to the ability to think the thought that bRa. In the case of inference, the ability to infer A&B from A&B&C is systematically related to the ability to infer only A from A&B&C. All this implies that mental representations, in general, have combinatorial syntax and semantics.

Fodor and Pylyshyn's argument can therefore be summarised as follows:

1. Human cognitive capacities are systematic.
2. The best explanation for the phenomenon of systematicity is to postulate the existence of mental states with combinatorial structure.
3. Connectionist architecture does not support representational states with combinatorial structure; only the classical architecture does.

Therefore, human cognitive architecture is classical.

We have seen why the first and second premises must be true. But what about the third? It is non-controversial that the Turing/ von Neumann cognitive architecture can support representational states with combinatorial structure. This, however, is not so in the case of the connectionist architecture. Connectionist networks, if they do not implement the classical architecture, are incapable of providing many of the basic operations necessary to process complex expressions. Here are some of the barriers to the processing of complex structures by connectionist architectures that have been widely recognized in the literature:

- Connectionist architectures are unable to provide consistent part/ whole relationships between hierarchical data structures (Fodor and Pylyshyn, 1988; Hinton, 1991).
- Connectionist architectures are unable to provide distal access to data structures (Touretzky, 1991). (This is also known as the symbol transportability problem.)
- Connectionist architectures are unable to provide the basic operation of binding a variable (Smolensky, 1991; Touretzky, 1991).
- Connectionist architectures are incapable of doing recursion (Fodor and Pylyshyn, 1988; Pollack, 1991).

Taken either jointly or separately the above considerations seem to warrant the belief that connectionist architecture cannot support representational states with

combinatorial structure. This, of course, should seal the fate of connectionism as a viable challenge to the classical approach to cognition. It is therefore understandable that connectionists have exerted much effort toward undermining the third premise of the systematicity argument. But would a successful challenge to this premise constitute a victory for connectionism?

Connectionism is incompatible with the classical approach only in so far as it is offered as an alternative representational theory of the mind, i.e. only in so far as it is offered as a psychological theory. Yet, it is important to point out that Fodor and Pylyshyn do not contest the possibility that connectionist networks could be used to model nonrepresentational states of an organism; by its very nature such an account would not have any significance for psychology because such an account could provide at best only a theory of the <u>implementation</u> of the classical architecture (Fodor and Pylyshyn, 1988, pp. 10-11). In other words, if connectionism provides a theory of the nonrepresentational neurological states of an organism that could be used to explain how the classical architecture is implemented in neural hardware, such a theory will not have direct relevance for cognitive psychology because it will have nothing to say on the question of which representations determine the behaviour of the organism. So, if connectionism is merely intended to provide a theory of the implementation of the Turing/ von Neumann

cognitive architecture, it is not strictly speaking incompatible with the classical view; it is simply not a theory about cognitive architecture.

This poses a very difficult dilemma for connectionism. If connectionist architecture cannot support complex mental states, connectionism is clearly inadequate as a psychological theory; if it <u>can</u> support complex mental states, it merely provides an implementation theory for the classical architecture. But then connectionism has no relevance for cognitive psychology. Connectionism is directly threatened by this inadequacy/ irrelevance dilemma. If connectionists succeed in answering the systematicity argument by providing a theory of the implementation of the classical architecture, they must forgo any claim to the relevance of connectionism for psychology. So the task facing connectionism is to show that connectionist architectures <u>can</u> process representational states with a combinatorial syntax and semantics. At the same time, connectionists must show that such architectures are <u>not</u> "mere" implementations of the Turing/ von Neumann architecture.

In marked contrast to productivity arguments, most connectionist openly or tacitly accept that human cognitive capacities are systematic. At the very least there are no connectionist arguments to the contrary. Many connectionists (for example Smolensky, 1987, 1991, Hinton, 1991, Touretzky, 1991, Pollack, 1991) also accept that in the absence of a

plausible alternative theory, postulating mental representations with combinatorial structure is the only way to account for the phenomenon of systematicity. They also credit Fodor and Pylyshyn's criticism for pointing out the importance of compositional representations for the representational theory of the mind. Speaking on behalf of this "silent connectionist majority" Smolensky has admitted that

> Until recently we have not had any systematic ideas about how to represent complex structures. In fact, it was Fodor and Pylyshyn who got me thinking about this, and ultimately convinced me [of the importance of representing complex structures]. (Smolensky, 1987, p.156.)

But if connectionists accept systematicity as an explanandum, and if they agree that in order to explain systematicity one needs mental representations with combinatorial structure, how can they escape the inadequacy/ implementation dilemma? Not surprisingly, connectionists have argued that this is a false dilemma. In a series of three articles Smolensky (1987, 1988, 1991) has tried to present a connectionist theory of the compositional character of mental representations that at the same time avoids the charge of "mere implementationism".

In order to avoid the "mere implementation" charge

Smolensky proposes to redefine the notion of an implementation. He correctly points out that to a large extent the notion of implementation, as it is used in cognitive science, is inherited from computer science. In computer science the implementation of one system by another means exact simulation of the behaviour of the first system by the behaviour of the second. As Smolensky puts it:

> If there is an account of a computational system at one level and an account at a lower level, then the lower one is an implementation of the higher one if and only if the higher description is a complete, precise, algorithmic account of the behaviour of that system. (Smolensky, 1990, p.203.)

The emphasis on exactness is all important here. Mere implementationism threatens connectionist theories only if implementation is taken to mean exact simulation. If on the contrary, this exactness condition is relaxed, so that connectionism is taken to provide only an "approximate implementation" of certain basic characteristics of symbolic architecture, then it is possible to claim that the roles should be reversed: rather than being a "mere implementation", connectionist architecture can be seen as a "refinement" of the classical architecture. As a result of this reversal, the Turing/ von Neumann architecture can be claimed to be only a crude approximation to the "real" cognitive architecture.

Accordingly, connectionists argue that the true cognitive level of description lies at the level of micro-features, activation patterns and connection strengths, rather than at the symbolic level of the classical architecture. Smolensky is only too happy to provide historical precedents to this reversal of roles -- Kepler's laws vs. Newton's; Classical mechanics vs. quantum mechanics -- in all cases symbolic theories of cognition are cast into the role of historically superseded less refined approximations. Thus, instead of trying to show that connectionism can have some significance for psychology, Smolensky undermines Fodor and Pylyshyn's criticism by simply assuming that connectionism <u>is</u> psychologically significant, while the classical architecture is at best only a crude approximation to the connectionist architecture. But there is a major problem with this shifting of the burden of proof: Classical architecture was adopted because it provided an explanation of such cognitive phenomena as productivity and systematicity. Can the connectionist "refinement" of the classical architecture do better in explaining these pervasive facts of cognition? Smolensky thinks it can and offers two connectionist theories which, he claims, can better account for the compositionality of representational states and thus better explain systematicity. Following Smolensky, we call these theories <u>weak</u> compositionality and <u>strong</u> compositionality.

Weak Compositionality

According to the theory of weak compositionality, thoughts must have a composite structure and mental processes are sensitive to this structure. Smolensky claims that many of the charges against connectionist architecture are based on a notion of connectionist representation that is too narrow. For example in their criticism of connectionism Fodor and Pylyshyn used as an example a particular network designed for automated resolution theorem proving (Ballard and Hayes, 1984) where each node of the network is labelled with the name of a single variable. Smolensky thinks that Fodor and Pylyshyn's criticism regarding compositionality is justified with respect to this specific connectionist model, as well as other "ultralocal" models in which each individual node is thought to represent a complete feature. However, Smolensky claims that the most promising connectionist models are the ones that use distributed representations, i.e. networks whose nodes are labelled with "microfeatures" instead of "macrofeatures" (Smolensky, 1988).

What is the difference between a macrofeature and a microfeature, between an ultralocal and distributed representation? Smolensky does not provide a firm criterion for distinguishing between them; all he does is to give a few suggestive examples. The only thing that seems to make one representation a distributed representation is the fact that the label of more than one node is considered to be part of

the representation. For example, a network which has one of its nodes labelled "coffee" apparently should be considered as giving an ultralocal representation of coffee, but according to Smolensky, a distributed representation of "coffee" might look like the representation in Figure 1.1. (See Figure 1.1.)

The reason why Smolensky thinks that distributed representations can be the answer to the systematicity/ compositionality argument is that this distributed representation of "coffee" was produced by subtracting the distributed representation of "cup without coffee" in Figure 1.2 from the distributed representation of "cup with coffee" in Figure 1.3. (See Figure 1.2 and 1.3.)

Smolensky admits that the operations of vector subtraction and vector addition are unlikely to satisfy the requirement of strict compositionality which is satisfied by a "classical" representation of cup with coffee. Weak compositionality only approximately resembles the combinatorial structure of mental representations as required by the classical architecture:

> ...the compositional structure is there, but it's there in an approximate sense. It's not equivalent to taking a context-independent representation of coffee and a context-independent representation of cup -- and certainly not equivalent to taking a context-independent

Figure 1.1

Representation of "coffee".

Units	Microfeatures
0	upright container
1	hot liquid
0	glass contacting wood
0	porcelain curved surface
1	burned odour
1	brown liquid contacting porcelain
0	porcelain curved surface
0	oblong silver object
0	finger-sized handle
1	brown liquid with curved sides and bottom

Figure 1.2

Representation of "cup without coffee".

Units	Microfeatures
1	upright container
0	hot liquid
0	glass contacting wood
1	porcelain curved surface
0	burned odour
0	brown liquid contacting porcelain
1	porcelain curved surface
0	oblong silver object
1	finger-sized handle
0	brown liquid with curved sides and bottom

Figure 1.3

Representation of "cup with coffee".

Units	Microfeatures
1	upri[illegible]t container
1	hot liquid
0	glass contacting wood
1	porcelain curved surface
1	burned odour
1	brown liquid contacting porcelain
1	porcelain curved surface
0	oblong silver object
1	finger-sized handle
1	brown liquid with curved sides and bottom

> representation of the relationship in or with - and sticking them together in a symbolic structure, concatenating them together to form the kind of syntactic compositional structure like with(cup, coffee) that Fodor and Pylyshyn want connectionist nets to implement (Smolensky, 1990, p. 208).

Smolensky's own admission shows why weak compositionality is, indeed, too weak to qualify as a solution to the compositionality problem for connectionist architectures. As Fodor and McLaughlin (1990, pp. 193-95) have remarked, the main problem here is the context-dependency of the distributed representations. While on the classical account the meaning of the constituent parts of an expression determines the meaning of the whole, in Smolensky's example CUP, COFFEE and WITH are not independent constituents whose meaning contributes to the meaning of the whole. The "constituents" of the complex of microfeatures CUP-WITH-COFFEE are also complexes of microfeatures - CUP-WITHOUT-COFFEE and COFFEE. But the representation of the meaning of the whole and of the parts is entirely context-dependent. For example, the representations of COFFEE produced from the contextual wholes of, say CAN-WITH-COFFEE or HALF-EMPTY-CAN-WITH-COFFEE, or HALF-EMPTY-CAN-WITH-COLD-COFFEE will have nothing in common. Indeed, to show weak compositionality for every possible coffee context we have to create a new type of distributed

representation to represent every new context. But there will be no systematic relationship between the "coffees" in these different contexts. This is a far cry from the part/ whole compositionality which, on the classical approach, is required to explain systematicity.

The context embededness of connectionist representations is a major obstacle to distal access relationships between representations. This problem appears to be recognized by Smolensky as well:

> But, one might well argue, the sense in which the vector encoding the distributed representation of cup with coffee has constituent vectors representing cup and coffee is too weak to serve all the uses of constituent structure -- in particular, too weak to support formal inference -- because the vector representing cup cannot fill multiple structural roles. (Smolensky, 1990, p.212.)

Thus Smolensky in effect concedes that weak compositionality cannot be considered a serious contender for the explanation of systematicity.

Strong Compositionality

Smolensky's first attempt to answer the systematicity argument failed because of the context dependence of "weakly compositional" representations. He tacitly agrees with such an assessment when he recognizes that the two main problems that plague most connectionist architectures are the distal

access problem and the variable binding problem (Smolensky, 1991, p. 160). The distal access problem (transportability problem) arises because of the immovability of the constituents of connectionist representations -- connectionist representations are always localized as labels to certain nodes of a network, and since the nodes cannot move around the network, connectionist representations are always grounded in a certain region of a network. In the same way a complex connectionist representation is just a localised region of the network that does not have access to the parts of the network which represent its constituents (cf. Smolensky, 1991, p.163). These constituents are locked in their corresponding regions of the network and cannot take part in other complex representations as required by the principle of compositionality of mental representations. It is clear that unless connectionist representations have transportable parts, connectionist architectures will be unable to meet the challenge of the systematicity argument. The eventual solution of this problem has been recognised by many connectionists as crucial to the success of their program (Touretzky, 1991; Pollack, 1991).

Closely connected with transportability is the variable binding problem. In most connectionist simulations the binding of values to a variable is entirely arbitrary. Certain regions of a network are simply labelled with the name of a variable and the patterns of activation are thought of as providing

values for these variables -- for example a certain node can be labelled as "made of porcelain" and the values can be "yes" or "no", represented respectively as activation or as absence of activation of the same node. In this way the label of the node serves as the variable, while the activation or non-activation of the node serves as the value of this variable. But such labelling is not a well defined encoding/ decoding operation and is in fact entirely arbitrary -- in our example, we could label the node as "made of wood", "made of metal", or anything we like; the activations of the node will still provide the values for these variables; the trouble is that we cannot be sure which variable the network is representing a value of. As a result connectionist architectures lack a well defined operations of binding and unbinding of a value to a variable (cf. Smolensky, p. 160). This is a severe limitation on connectionist systems because without the binding and unbinding operations they cannot create and maintain representations with constituent structure.

In order to solve the distal access (transportability) and the variable binding problem, without at the same time giving in to implementationism, Smolensky introduces a new class of connectionist representations -- tensor product representations.

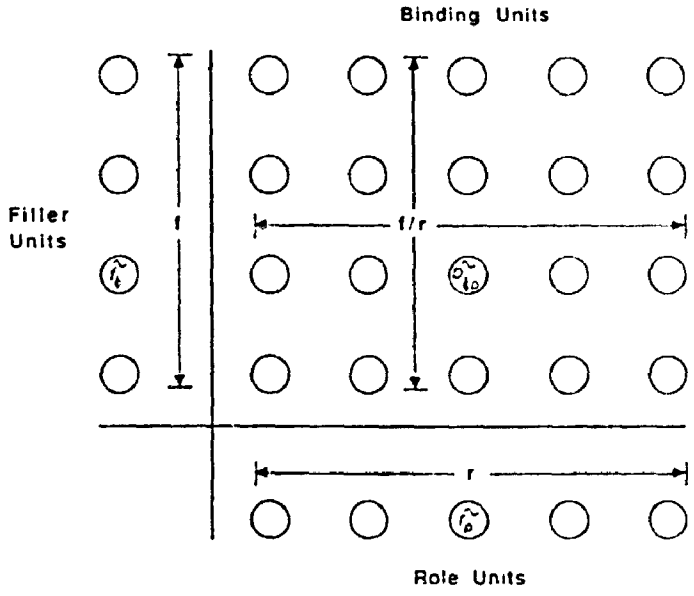
A tensor product representation is constructed in the following way: The nodes of a network (see Figure 1.4) are divided in three parts -- "filler" units (units encoding

Figure 1.4

The tensor product representation

for filler/ role bindings.

(Reprinted from Smolensky, 1991.


Binding Units
Filler Units
f
f/r
r
Role Units

values), "role" units (units encoding functions), and "binding" units (encoding the function of a given value). The vectors of positive activations over the set of filler units represents the "atomic" constituents such as phonemes, letters, words, etc. The vectors of activations over the role units represent the roles these constituents can have in more complex representations such as nasal(phoneme), first(), second(), etc. (letter in a word), or subject_of(word), complement_of(word), etc.

The network in question has four fillers -- the letters H,J,N,O and four roles First(letter in a 4-letter word), Second(), Third(), and Fourth(). Each filler and each role are represented by a four component vector as follows:

H	J	N	O	First	Second	Third	Fourth
$f_1 = \begin{matrix}1\\0\\0\\0\end{matrix}$	$f_2 = \begin{matrix}0\\1\\0\\0\end{matrix}$	$f_3 = \begin{matrix}0\\0\\1\\0\end{matrix}$	$f_4 = \begin{matrix}0\\0\\0\\1\end{matrix}$	$r_1 = \begin{matrix}1\\0\\0\\0\end{matrix}$	$r_2 = \begin{matrix}0\\1\\0\\0\end{matrix}$	$r_3 = \begin{matrix}0\\0\\1\\0\end{matrix}$	$r_4 = \begin{matrix}0\\0\\0\\1\end{matrix}$

If we want to represent a filler as bound to a role, e.g. that the letter J is the first letter of a four letter word, we simply take the outer product of the vectors f_2 and r_1 (we transpose the second vector and multiply each f_{2i} with each r_{1j}) to produce a matrix W with i rows and j columns:

$$f_2 \times r_1 = W_{ij} = \begin{matrix}0&0&0&0\\1&0&0&0\\0&0&0&0\\0&0&0&0\end{matrix}$$

Representation of 'First(J)'

Since a vector can be regarded as a tensor of rank one and a matrix as a tensor of rank two, we can say that J, as a first letter in a four letter word, is represented by the tensor (of rank two) W_{ij}. In the same way we can represent Second(O), Third(H), and Fourth(N) as outer products of their corresponding vectors or as tensors of rank two. (Eventually, more complex representations with tensors of higher rank can be built.)

According to Smolensky tensor product representations are best suited for representing arbitrary fillers (values) as bound to roles (functions). The main advantage of these connectionist representations lies in their apparent ability to represent true constituents which may fill different roles in different complex representations, thus solving the transportability problem as well.

Smolensky claims several additional advantages for this new type of connectionist representation: (i) the tensor product operation is potentially much more powerful than vector addition and subtraction (used in the account of weak compositionality). (ii) These new representations can use continuous as well as discrete activation values. (iii) The tensor product representations are not context dependent. According to Smolensky, this is the most notable departure from weak compositionality (1991, pp.163-64). However, there are still some very difficult problems with this account of compositionality. The first has to do with the <u>reversibility</u>

of binding operations on these connectionist representations. For example, in order to bind a filler to a role, for example 'J', to 'First()', we have to produce the outer product of the vectors f_2 and r_1 (rank two tensor W_{ij}). But is there a way to produce f_2 from W_{ij}?

The problem with the "unbinding" of f_2 from its outer product with r_i is that this operation is guaranteed to succeed only as long as the vectors representing the different roles are linearly independent as in the above example. The same applies to the vectors representing the fillers. If they are not linearly independent there is no guarantee that the unbinding operation will succeed. For example, from the representation of First(J) we might end up unbinding the representation of H, or N, or O. Unlike the classical architecture where operations on representations are in principle reversible, tensor product binding of variables does not guarantee their successful subsequent retrieval in all cases where fillers and roles are coded by linearly dependent vectors.

Smolensky is aware of how damaging the unbinding problem is for strong compositionality. He offers to distinguish between exact unbinding procedures and self-addressing unbinding procedures. The first procedure is defined only for linearly independent cases and is guaranteed to succeed; the second procedure is defined for all cases but is not guaranteed to result in correct unbindings. The trouble is

that exact unbinding comes at a very steep price. As Smolensky himself admits

> In order to guarantee faithfulness of representations, it will often be necessary to impose the restriction of linearly independent patterns for the constituents. This restriction is an expensive one, however, since to have n linearly independent patterns one needs to have at least n nodes in the network. (Smolensky, 1991, p. 172.)

It should be obvious that the linearity restrictions for the exact unbinding procedure are too restrictive. They imply a rigid upper bound on the complexity of connectionist representations. But we do not find such artificial bounds in human cognition. Certainly, it would be absolutely wrong to suggest that the expansion of our cognitive capacities is possible only through the addition of more neurons in the brain.

The unbinding problem is by no means the only problem that puts in question Smolensky's tensor product representations. We saw that strong compositionality was offered as a possible solution to the context-dependency of connectionist representations. However, context-dependency is not entirely overcome with the help of tensor product representations. Even if we set aside the unbinding problem for the moment and assume that the binding/ unbinding operations are unproblematic, it is not difficult to see that

tensor product representations are still dependent on the choice of a particular network with a particular capacity to represent a given number of variables and a given number of roles. For example, let us suppose that the network in Figure 1.4 can represent the "atomic" constituents H,J,N,O as well as the structured "molecular" expressions JOHN, HNJO, JHNO, etc. This fact however, does not mean that these atomic constituents can take part in other more complex representations, like JANEJOHNES for example. In order to insure this possibility we have to build a new network with units for three more atomic components -- A,E,S, and six more role units. But the representations of let us say J in the new network will not have anything in common with the representation of J in the old network -- it will be a completely different tensor. The same is true of any other larger network. It is clear that the compositionality that might be exhibited by a tensor product representation is strictly dependent on a particular choice of roles and variables. This role/ variable context acts as an <u>external limit</u> to contacts with other representational structures. In particular, tensor product representations found in one filler/ role context cannot be constituents of representations found in other filler/ role contexts. But it is quite unreasonable to assume that a network anticipating all possible contexts can be built. Clearly, the <u>context problem</u> that undermined the theory of weak compositionality has

resurfaced again.

In addition to the binding and the context-dependency problem there is a third major weakness in Smolensky's account of strong compositionality. Fodor and McLaughlin (1990, p.199) point out that a complete solution to the systematicity problem should be able to account for not only the semantic interpretability of complex mental representations but also for their causal roles, including the causal roles of their constituents.

In classical architectures complex mental representations are causally sensitive to constituent structure, i.e. their components can be causally efficacious. This however is not the case with tensor product representations. As we saw, when a tensor representing a particular filler/ role binding is tokened, a particular pattern of activations is present in the network. It is this tensor (or the corresponding activation pattern) that alone has a causal role. The constituent tensors -- the filler and role vectors -- are not tokened, i.e. their units are not activated. But this means that they are not causally efficacious. The compositionality of the vector product representation is, so to say, only in the eye of the beholder, it is not causally present in these connectionist representations.

Conclusion

To summarize: I have argued that the failure of connectionism as a cognitive theory stems from its inability to explain the phenomena of productivity and systematicity. The inability of connectionist architectures to support semantically interpretable states with complex constituent structure implies that human cognitive architecture cannot be connectionist. Therefore, connectionist ANNs cannot serve as finitary models of human cognizers.

CHAPTER TWO

On the Symbolic/ Subsymbolic Distinction

Introduction

The advent of the new connectionism has sparked a far reaching debate on the nature of human cognitive architecture. In a wide ranging attack on the basic tenets of the new connectionism, Fodor and Pylyshyn (1988) argued that connectionism cannot offer an alternative to the classical Turing/ von Neumann cognitive architecture; at best, connectionist systems can serve as possible models for implementing the classical architecture in neural hardware. Paul Smolensky has challenged Fodor and Pylyshyn's criticisms, and with his "proper treatment of connectionism" has tried to establish connectionism as a new "paradigm" in cognitive science (Smolensky, 1987, 1988). An integral part of Smolensky's defense of connectionism is the introduction of the symbolic/ subsymbolic distinction.

In what follows I will show that there are several symbolic systems that are perfectly capable of carrying out what Smolensky calls "subsymbolic computation", and that, therefore, Smolensky's introduction of this distinction is entirely spurious.

The Reasons for the Introduction of the Symbolic/ Subsymbolic Distinction

Smolensky (1988) has rejected what he calls the symbolic paradigm in AI, i.e. the view that virtually all cognitive tasks can be simulated by programs consisting of linguistically formalized rules that are sequentially interpreted. The main reason for his rejection of the symbolic paradigm is that it has led to a number of disappointing results (Smolensky, 1988, p.5):

> [A1] Actual AI systems [based on the symbolic paradigm] seem too brittle, too inflexible, to model true human expertise.
>
> [A2] The process of articulating expert knowledge in rules seems impractical for many important domains (e.g. commonsense [knowledge]).
>
> [A3] [The symbolic paradigm] has contributed essentially no insight into how knowledge is represented in the brain.

Smolensky proposes to replace the symbolic paradigm with what he calls the subsymbolic paradigm. As a first approximation, the subsymbolic paradigm is the view that human cognitive architecture is connectionist and that connectionist cognitive architecture defines a new subsymbolic level of computation.

But what are the determining properties of this new subsymbolic level of computation and of the new subsymbolic paradigm? According to Smolensky,

> The name "subsymbolic paradigm" is intended to suggest descriptions built up of entities that correspond to constituents of symbols used in the symbolic paradigm; these fine-grained constituents could be called subsymbols, and they are the activities of individual processing units in connectionist networks. Entities that are typically represented in the symbolic paradigm by symbols are typically represented in the subsymbolic paradigm by a large number of subsymbols. Along with this semantic distinction comes a syntactic distinction. Subsymbols are not operated upon by symbol manipulation: They participate in numerical - not symbolic - computation. (Smolensky, 1988, p.3.)

The key to understanding the difference between symbolic and subsymbolic representation is the difference between "quasilinguistic representations" and "good old-fashioned numerical vectors" (Smolensky, 1988, p.5). According to Smolensky only connectionist neural networks can represent (fine-grained) knowledge with numerical vectors, i.e. vectors of the form $\underline{v}=\langle v_1, v_2, \ldots, v_n \rangle$ where each v_i represents a "microfeature" of one or many "macroconcepts". Symbolic systems are limited to representing knowledge only quasilinguistically. Accordingly, a symbolic system will be limited to representing only the individual symbols, and can represent no microfeatures whatsoever.

Smolensky claims that connectionist systems that are capable of using subsymbolic computation can solve the "hardness" problem (A1) faced by the symbolic systems (Smolensky, 1987). Since the individual neurons do not encode concepts, conceptual knowledge is represented by complex patterns of activations over many neurons. The interaction between these activity patterns is not directly described by a formal definition, but can be computed only approximately. Smolensky insists that

> ...there will generally be no precisely valid, complete, computable formal principles at the conceptual level; such principles exist only at the level of individual units - the subconceptual level (Smolensky, 1988, p.3).

The ability to perform approximate computations and thus to display "soft" behaviour is, in Smolensky's opinion, the real advantage of subsymbolic systems over symbolic systems. According to Smolensky, symbolic systems are 'hard' and are in principle unable to display 'soft' behaviour. He rejects the possibility that any kind of softness can emerge out of the hard rules of symbolic AI systems (Smolensky, 1987, p.137).

In summary, the subsymbolic paradigm is defined by the following principal claims:

B1. Subsymbolic systems -- unlike symbolic systems -- can represent concepts as sets of subconcepts or

microfeatures; this is because subsymbolic systems encode concepts as numerical state vectors.

B2. Subsymbolic systems -- unlike symbolic systems -- are capable of soft or approximate computations which imply the ability to handle noisy, incomplete data, and the ability to display graceful degradation of performance.

B3. Artificial neural networks (ANNs) that carry out subsymbolic computations -- unlike symbolic systems -- are very much like the brain.

In what follows I will try to show that the proposed subsymbolic level of computation is entirely spurious. I will argue that the ability to handle microfeatures and vector representations is by no means the exclusive property of connectionist systems. Symbolic systems can make very good use of vector representations with equally convincing or even better results. I will show that the symbolic paradigm can produce systems that are not brittle, that are very flexible, that can model true human expertise, and that can actually do better than many human experts. I will show that such systems can articulate human knowledge in rules in surprisingly many domains, including what are commonsense domains. In addition, I will show that connectionist learning algorithms have contributed essentially no insight into how knowledge is represented in the brain.

The Brain-Likeness of Error Backpropagation in ANNs

I will begin in reverse order with an examination of the brain-likeness of connectionist architecture. There can be no doubt that the great appeal of the ANNs is largely a result of their brain-like appearance. But the question of the extent to which they can be viewed as models of the brain is rarely addressed.

Smolensky agrees that the invective "Look how the brain does it and do the same!" is not likely to be very useful in the near future. He also rejects the idea that connectionist models are exact models of the brain, but he claims that connectionist architecture "abstractly models a few of the most general features of neural networks [of the brain]" (Smolensky, 1988, p.6).

The phrase "abstractly models" can be abused so that one can dismiss almost any counterexample as not affecting the abstract brain-likeness of connectionist systems. In order to avoid such an evasion I will concentrate on what is undeniably the main learning algorithm used in most connectionist systems - error backpropagation (BP).

Backpropagation in ANNs (or the Generalized Delta Rule (GDR), as this algorithm is also known) is a very interesting learning algorithm. The task is to learn to associate a set of input and output patterns, so that after the supervised training process is completed, the network when given an input pattern from the set, will produce its corresponding output

pattern. The training is carried out in the following way:

> The ANN (with usually randomized weights) first uses the input vector which is propagated forward to produce its own output vector. Then it compares its output with the target vector (this is the pattern it has to learn). If the output vector and the target vector are the same, training stops. If they are different, the difference is reported as an error signal for each output unit. Then the error signal is passed backwards to each unit in the network and the appropriate weight-changes are made. The forward and backward procedures form one epoch. Such epochs are repeated until the output vector is identical with the target vector (or until the difference is minimized to a desired level). (Cf. Rumelhart, Hinton, and Williams, 1986, p. 327.)

After an ANN has been trained, it can output the target pattern when presented with the original one. We then can say that the network has learned to associate the two patterns.

It should be quite obvious that nothing remotely similar to error backpropagation is happening in the brain. Actual biological neurons are not involved in passing of <u>error messages</u> back to where the stimulation has come from. Patterns of neural firing do not constitute <u>epochs</u> of forward/ backward passing of electrical impulses. In any case the "units" whose weights are changing during backpropagation are not intended

to model -- even remotely -- real neurons. Most connectionist researchers would agree with the statement that

> These units often have properties similar in some respects to neurons ... However, their inventors [should] always be careful to point out that they are not intended to represent real neurons. Indeed, at this stage of the game, it would be foolish to attempt to do this. (Crick and Asanuma, 1986, pp. 396-7.)

This is not to say that BP is a useless algorithm. On the contrary, it is a very interesting and potentially a very powerful learning algorithm. My point here is that it has nothing to do with the actual workings of the brain. The fact of the matter is that connectionist systems using BP are not more brain-like than symbolic systems which refuse to speculate on implementational issues; they are simply unlike the brain. Therefore, B3 is false (at least with respect to BP) and I think it is only fair to rewrite A3 as

> A3*. Connectionist systems, as well as symbolic systems which make no commitments as to their exact implementation in the brain, have contributed essentially no insight into how knowledge is represented in the brain.

The Problem of Brittleness

The brittleness of some AI systems has long been identified as one of the main bottlenecks in AI research. A program may work well for ideal cases but when confronted with real-world data that are often incomplete or noisy it breaks down. What ideally we would like to see is an undiminished or only gracefully degraded level of performance in the presence of noise. Inflexibility in such cases is obviously a severe handicap if one tries to model actual human expertise. It is true that some AI systems are very brittle in this sense. But what is true of some is not necessarily true of all members of a class.

There are several learning systems that have quite successfully overcome both of these problems. For example the family of symbolic learning systems based on the Top-Down Induction of Decision Trees (TDIDT) are particularly interesting for our purposes because several such systems, most notably ID3 (Quinlan, 1986a), C4 (Quinlan, 1986b), CART (Breiman, Friedman, Olsen, and Stone, 1984), ASSISTANT (Cestnik, Kononenko, and Bratko, 1987), have overcome the problem of brittleness in many respects and compare well with the family of ANNs learning systems based on BP.

First, I will examine two of the TDIDT systems -- ID3 and C4 -- in detail, then I will look at several experimental tests of ID3 and C4 on real-world induction tasks. I will also discuss the results of two experiments comparing the

performance of ID3 and CART systems with that of ANNs using BP.

The ID3 Symbolic Learning System

ID3 has been designed by J.R.Quinlan (1979, 1983a). Its historical predecessor is Hunt and Marin's Concept Learning System (CLS). (Hunt, Marin, and Stone, 1966) A new and more advanced version of ID3 is C4 (Quinlan, 1986b). ID3 is suitable for a comparison with the backpropagation algorithm (ANNs using BP) because both share a set of common features:

- Both are members of the class of supervised learning systems.
- Both learn from examples.
- Neither is application specific. Examples of successful applications for ID3 are chess strategies, weather prediction, medical diagnosis, voting pattern predictions, credit card application assessment, etc. BP has also been used in a wide array of applications.
- Both are able to generalize from training examples that they have been exposed to during the learning process to new "unseen" cases.

The difference between ID3 and ANNs using backpropagation (BP) lies in the form of knowledge representation:

- ID3 represents the acquired knowledge as a decision tree. A decision tree can easily be converted into a set

of production rules. In most cases this form of knowledge representation is "transparent" to the human observer i.e. humans are able to understand and use these rules.

- ANNs using BP store the acquired knowledge in their weights. The knowledge stored in this way is completely "opaque" to the human observer.

To understand how the decision-tree building algorithm of ID3 works it is useful to look at a concrete example. ID3 is capable of supervised learning from examples, so ideally it should be able to acquire expert knowledge by learning from human experts. Consider the following case: Many newspaper and radio stations have experts who check the weather forecast for the day, the current temperature and humidity, and give a simple piece of advice: e.g. whether the day will be suitable, or unsuitable, for playing golf. Now, ID3 can be trained by providing it with a sample of the past decisions of such an expert and can successfully generalize from the examples it has been exposed to during learning to future cases.

In order to be trained, ID3 is provided with the same universe of objects and with the same information that is taken into consideration by the human expert. For this learning task, the universe will be the days of the week. Each day can be described with the following attributes:

outlook: with values, {sunny, overcast, rain}.

temperature: with values, {cool, mild, hot}, or

{continuous}.

humidity: with values, {high, normal}, or {continuous}.

windy: with values, (true, false}.

For example a particular object, say <Saturday, June 23, 1990> might have the these characteristics:

outlook: overcast

temperature: cool

values: normal

windy: false

Given such a description the expert has to decide whether the day will be suitable for playing golf or not. The task is to classify each object in the universe as belonging to one or other of two classes which we may call P (play golf) and N (do not play golf).

Of course, without a training-set of examples of expert decisions, ID3 will not be able to learn anything. So suppose we are to provide ID3 with the set of training examples in Table 2.1 (number 1 to 10) say of saturday mornings in the past few years together with the decisions of a human expert. (See Table 2.1.)

It is interesting to note that each object in the universe can be encoded as a four dimensional state vector, where each vector component v_i of $\underline{v}=<v_1, v_2, v_3, v_4>$ corresponds

Table 2.1

Golf Data Set

No.	Attributes				Decision
	Outlook	Temperature	Humidity	Windy	
1	sunny	hot	high	false	N
2	sunny	hot	high	true	N
3	overcast	hot	high	false	P
4	rain	mild	high	false	P
5	rain	cool	normal	false	P
6	rain	cool	normal	true	N
7	overcast	cool	normal	true	P
8	sunny	mild	high	false	N
9	sunny	cool	normal	false	P
10	rain	mild	normal	false	P
11*	sunny	mild	normal	true	P
12*	overcast	mild	high	true	P
13*	overcast	hot	normal	false	P
14*	rain	mild	high	true	N

Note. Adapted from "Induction of Decision Trees" by R. Quinlan, 1986, Machine Learning, Vol. 1, pp.81-106.

to the value of a different attribute. The fact that ID3 can

operate on knowledge represented via state vectors will turn out to be very important when we compare it to the connectionist systems using BP.

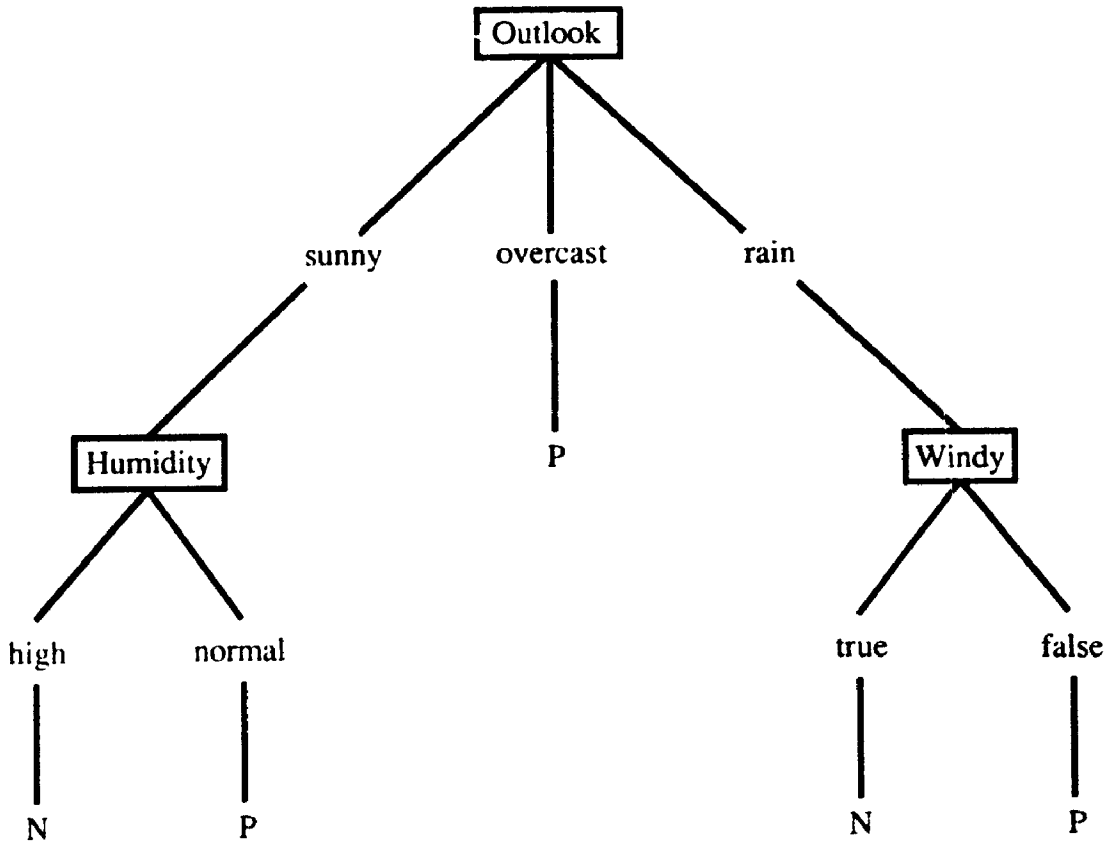
ID3 can extract the expert knowledge contained in this data by trying to build a decision tree that will classify the items in the training set (i.e. the first ten items). A simple tree that does the job can look like the one in Figure 2.1. (See Figure 2.1.)

ID3 builds the decision tree by employing a divide and conquer strategy. It first picks an attribute - in this case the attribute outlook - and checks its values. As it turns out all objects with value overcast for the attribute outlook belong to the play golf class, so ID3 closes this branch. In case a value does not classify a subset as belonging to only one class the search is extended by looking at another attribute and its values, until all subsets of objects are exhaustively classified.

We can simply check to see that this tree can correctly classify all training examples. Let's look at day #1. We start from the top node outlook; its value for #1 is sunny, so we choose to follow the leftmost branch. As a result, we arrive at the attribute humidity; its value for #1 is high and the literal under it is N. The decision tree has classified correctly the first item as belonging to the N class. In the

Figure 2.1

Decision tree for predicting weather conditions.


Outlook
sunny
overcast
rain
Humidity
P
Windy
high
normal
true
false
N
P
N
P

same way the decision tree can classify all training examples. But what is more interesting, it can generalize from the seen cases to the unseen cases # 11-14 and classify them correctly as well.

It is easy to see that this decision tree can immediately be converted into a set of production rules just by tracing all of its branches from the root to the terminal leafs. These rules will have the same predictive accuracy as the decision tree itself:

Rules produced from the tree on Fig. 2.1:

Rule 1: If outlook = overcast,
then play golf.

Rule 2: If outlook = sunny and humidity = normal,
then play golf.

Rule 3: If outlook = rain and windy = false,
then play golf.

rule 4: If outlook = sunny and humidity = high,
then do not play golf.

Rule 5: If outlook = rain and windy = true,
then do not play golf.

But how does the ID3 choose from which attribute to start building the decision tree, so that the simplest possible tree is created? For example, if ID3 chose the initial node to be the attribute humidity the resulting decision tree would be much more complex. In a rich domain with many attributes and

many values, choosing the wrong attribute can bring with it unnecessary computational costs. Ideally, the "blind" divide and conquer strategy has to be supplemented by some useful heuristics. ID3 uses a heuristic principle similar to Occam's razor: it uses an expected information gain criterion that chooses the "most relevant" attribute as the root of a (sub)tree, i.e. the attribute that reduces randomness in the remaining data as much as possible. This results in trees that generally branch out from the more informative to the less informative attributes, thus simplifying the overall tree structure.

How to Avoid Brittleness

ID3, C4, as well as some other members of the TDIDT family have to a large degree successfully overcome the problem of brittleness by augmenting the simple divide and conquer algorithm they all share. They are capable of dealing with a wide range of real-world problems, i.e. data sets that include noisy or partial information.

C4 has several features that enable it to deal with real-world data. One problem faced by the decision tree algorithms is that if the data contains contradictory information, i.e. noise, it may be prohibitive to continue subdividing the original set and its subsets until all members are classified. There is a need for a stopping criterion (Kononenko, Bratko, and Roskar, 1984) that would count n number of exceptions as

acceptable. This would stop the otherwise useless search. With different stopping criteria different levels of noise can be filtered out.

Another problem faced by C4 is the problem of missing data. A number of techniques have been suggested - when an attribute with a missing value is encountered during the tree building process there are a number of options: ignore the case with the missing value, "fill in" the missing value with the most common value for this attribute, etc. (Quinlan, 1989). The adoption of any of these solutions is likely to affect slightly the predictive success of the corresponding decision trees but overall any of these solutions would be able to deal with the missing data problem. It is remarkable that similar "filling in" techniques are used by the human visual system (the blind spot), by the human auditory system (the phoneme restoration effect), and by human memory (the phenomenon of constructive memory).

Table 2.2 summarizes some typical results testifying to the success of ID3 in solving real-world problems. (See Table 2.2.)

Table 2.2

ID3 Performance on Several Real-World Tasks

Data set	Number of attributes	Number of training/ testing ex.	Task	Human expert accuracy	ID3 accuracy
Sleep	13 numerical	3651/1824	classify stages of sleep	54%	72.6%
Mushrooms	22	5416/2708	classify poisonous or not	52%	99.9%
Futures prices	>10	337/169	predict up, down, or stable	38%	40.0%

Note. This table summarizes data reported in "Experiments on the Costs and Benefits of Windowing in ID3" by J. Wirth, and J. Catlett, 1988, J. Laird (Ed.) Proceedings of the Fifth International Conference on Machine Learning, pp.87-99, San Mateo, CA.

But probably the best example of the success of the decision tree algorithms in dealing with noisy, real-world data comes from several experiments with C4 on medical diagnosis. In these experiments the training data set came directly from hospital records. In one of them it consisted of 2800 hyperthyroidism cases from the Gavran Institute of Medical Research, Sydney. The data set had typical real-world characteristics: some of the values of attributes were unknown (up to 20% in some cases), some conditions such as secondary hyperthyroidism were represented by very few cases, worst of all some attribute values were incorrect. An additional difficulty was the fact that six of the attributes were numerically valued (reflecting results of hospital tests).

Despite the presence of contradictory information (noise) and despite the significant amount of missing data, C4 was able to induce a decision tree which correctly classified 99% of the unseen cases! (Quinlan, 1986b). In a later similar experiment the misclassified cases (e.g. the errors the program supposedly made) were submitted for review at the Gavran Institute. Surprisingly the review process discovered that in 9 of the 10 "errors" the mistake was actually committed by the human expert (Quinlan, Compton, Horn, and Lazarus, 1987, p.167). So, C4 was right after all!

Similar success in dealing with noisy real-world data is reported for the CART symbolic learning systems. Weiss and Kapouleas (1989) tested CART on iris classification data - a

standard data set used by statisticians. The CART tree was able to classify correctly 96% of the cases. They also tested CART on real-life diagnosis of appendicitis on 106 hospital records. CART performed roughly three times better than the human experts and diagnosed wrongly only 9.4% of all cases.

Impressive results have been reported about the pruned tree procedure of ASSISTANT. Kononenko, Bratko and Roskar (1986) tested ASSISTANT on actual cancer patient data and showed that ASSISTANT's accuracy was 72% compared with 64% for the expert physicians.

Such examples should convince even the sceptic that there exist symbolic systems which can handle noisy and incomplete information without breaking down. Therefore, A1 is false.

Commonsense

Smolensky claims that the process of articulating expert knowledge in rules seems impractical for many important domains (e.g. commonsense). <A3>. I think that this is an erroneous assumption and I will try to refute it by offering some fairly convincing counterexamples.

My example comes from politics. The data is drawn from the U.S. congressional voting records. The commonsense learning task is to identify the political affiliation of every congressman or congresswoman by looking at his or her voting pattern. C4 has to be able to learn how to tell a Democrat from a Republican by identifying the issues they

stand for or against. A similar learning task is faced by an adolescent who is to develop even a limited understanding of political life. So, if C4 successfully learns to distinguish correctly political affiliations, this surely should count as acquiring some commonsense knowledge.

In this experiment C4 had to process information organized in such a way as to include votes for each of the members of the U.S. House of Representatives on the 16 key issues identified by the Congressional Quarterly Almanac (Schlimmer, 1987). The CQA lists nine different types of votes: voted for, paired for, and announced for (these three simplified to yes), voted against, paired against, and announced against (these three simplified to no), voted present, voted present to avoid conflict of interest, and did not vote or otherwise make a position known (these three simplified to an unknown disposition).

There are 16 attributes corresponding to the 16 key issues which can take values {y[es], n[o], u[nknown]} and two major classes: democrat and republican. (See Table 2.3.) In the actual experiment C4 looked at 300 voting patterns creating a decision tree that correctly classifies 97% of the training examples. (See Figure 2.2.)

It is worth noting how economical this representation is. Instead of using all 16 attributes, the decision tree is constructed using only 9 of the most informative attributes. Moreover, C4 notices that most of the examples can actually

Table 2.3

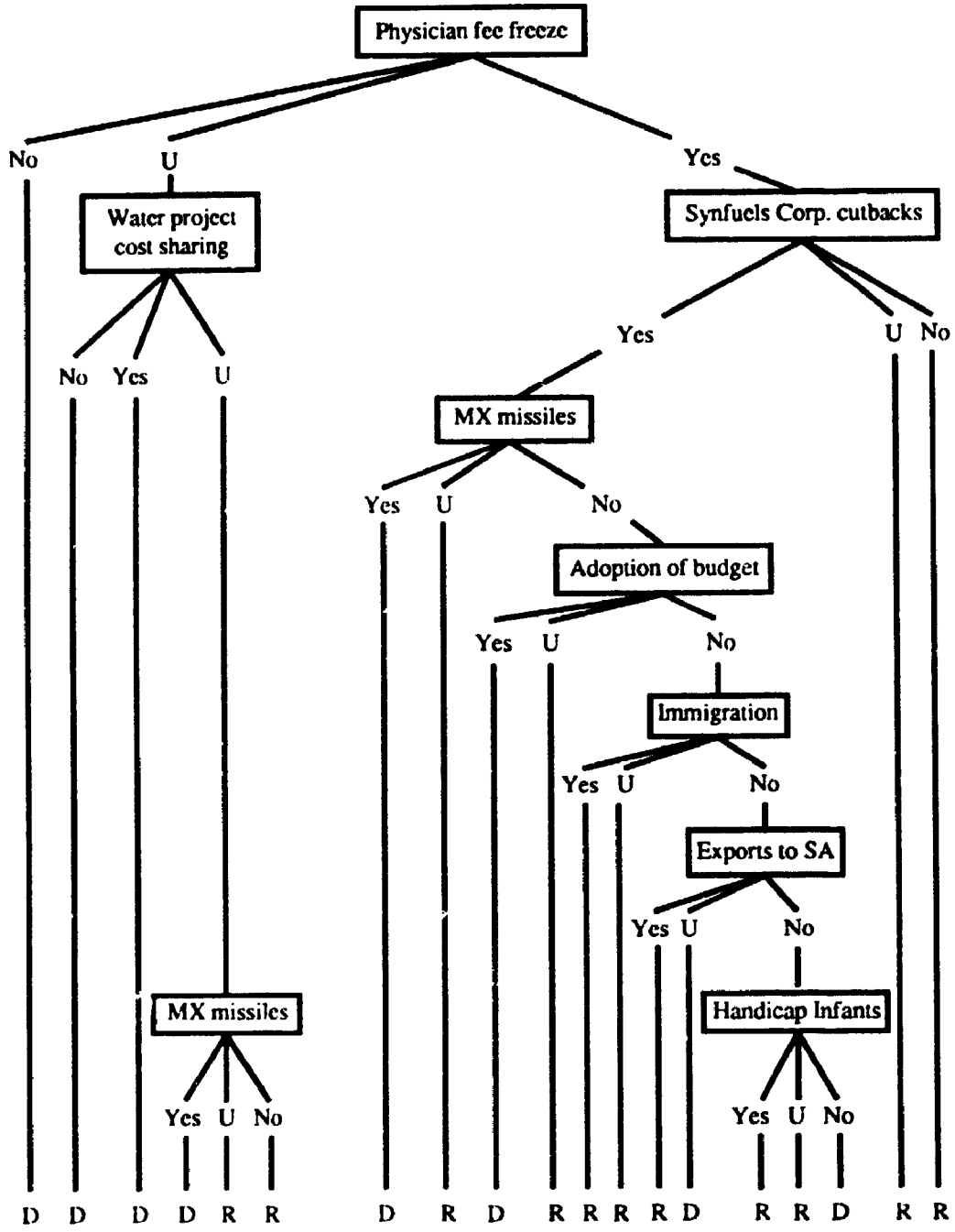
Votes on Proposed Legislation

Attributes	Values
Handicapped infants	n, y, u
Water project cost sharing	n, y, u
Adoption of the budget resolution	n, y, u
Physician fee freeze	n, y, u
El Salvador aid	n, y, u
Religious groups in schools	n, y, u
Anti satellite test ban	n, y, u
Aid to nicaraguan Contras	n, y, u
MX missiles	n, y, u
Immigration	n, y, u
Synfuels corporation cutbacks	n, y, u
Education spending	n, y, u
Superfund right to sue	n, y, u
Crime	n, y, u
Duty free exports	n, y, u
Export administration act South Africa	n, y, u

Note. Adapted from "Vote" [Machine-readable data file] by J. Schlimmer,1987.

Figure 2.2

Decision tree for predicting party affiliations.


Physician fee freeze
No
U
Yes
Water project cost sharing
Synfuels Corp. cutbacks
No Yes U
Yes
U No
MX missiles
Yes U No
Adoption of budget
Yes U No
Immigration
Yes U No
Exports to SA
Yes U No
MX missiles
Handicap Infants
Yes U No
Yes U No
D D D D R R
D R D R R R R D R R D R R

be classified by pruning the original tree to the Physician fee freeze attribute. The branches of this pruned tree can successfully classify 95% of the training data. (See Figure 2.3.)

The predictive power of both the overgrown and the pruned decision trees is quite surprising. When tested on 135 unseen examples the first tree classified 98.5% and the second 97% of all unseen examples. This is an error rate of 1.5% and 3% respectively! This shows how successful C4 can be in generalizing from past to future cases.

Of course there is nothing mysterious about the way the decision trees acquire and represent knowledge. Unlike the opaque representation of knowledge at the subsymbolic level, each sequence of branches of a decision tree can be converted into a production rule. For example one fairly complicated production rule might be:

physician fee freeze = y & synfuel corp. cutbacks = y &
MX missiles = n & adoption of budget resolution = y
--> class: democrat

But even fairly simple production rules like

Rule 1:

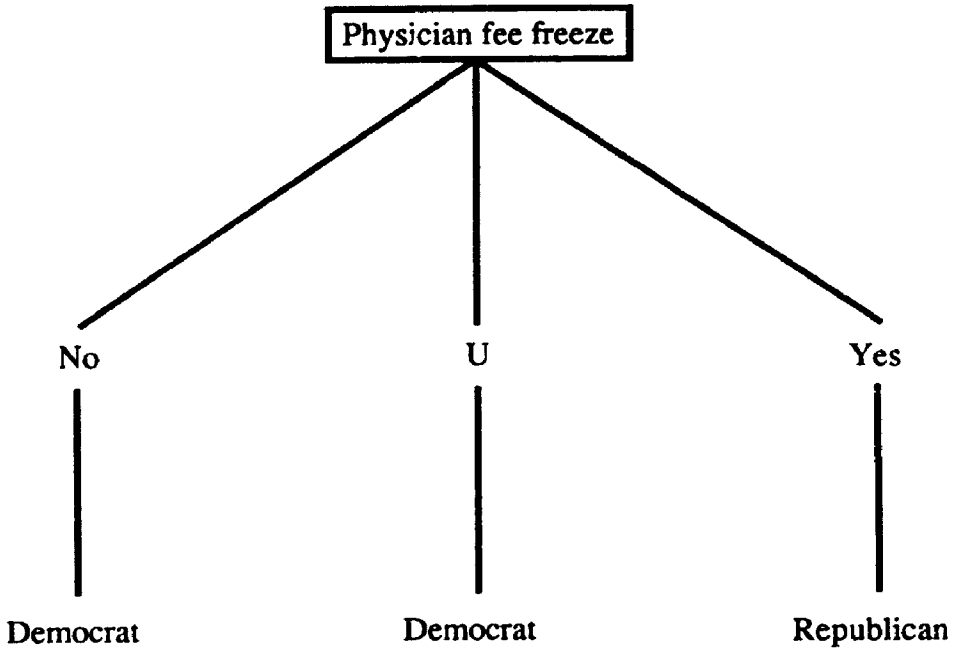
physician fee freeze = n
-> class: democrat

Rule 2:

physician fee freeze = y
-> class: republican

Figure 2.3

Pruned decision tree for predicting party affiliations.


Physician fee freeze
No
U
Yes
Democrat
Democrat
Republican

are able to classify correctly almost all of the unseen cases with error rates of 1.3% and 5.6% respectively.

If we examine these simple production rules derived from the pruned tree we can see that they actually represent very useful rules of thumb which have the typical look and feel of commonsense knowledge. These rules in effect are overgeneralizations - e.g. they misclassify a small number of examples - but so are most commonsense rules of thumb. Both commonsense rules and production rules from pruned decision trees share a certain degree of imprecision but they are both surprisingly useful in many everyday domains.

It is a pity that authors like Smolensky often give A1-A3 as their standard justification for why we need connectionist systems. ANNs using BP have enough interesting features to merit a detailed investigation without such unhelpful justification.

On the Notion of Subsymbolic Computation

It is one thing to show that many symbolic learning algorithms do not suffer from the ailments diagnosed by Smolensky but it is quite another to show that there is no principled distinction between symbolic and "subsymbolic" levels of representation, that ANNs are not unique in their ability to work with "subsymbols", and that there is no incompatibility between the symbolic and the subsymbolic paradigms.

In order to demonstrate the spuriousness of the symbolic/subsymbolic distinction we have to compare the performance of ANNs and symbolic systems solving one and the same problem on one and the same data. If, contrary to Smolensky's predictions, it turns out that there are no systematic differences in the performance of ANNs and symbolic programs in solving one and the same learning problem on one and the same body of data, then we will have a very good reason to believe that the two systems are not fundamentally incompatible with respect to their learning and generalization capacities.

Luckily, the AI community has not been idle since the publication of PTC in 1988 and if at the time Smolensky wrote his article there were almost no comparative tests of the computational abilities of ANNs with other systems, now such experimental studies are widely available.

One very interesting experimental comparison of BP with ID3 learning algorithms was carried out by Shavlik, Mooney and Towell (1991). They compared the performance of ID3 and ANN using BP on five standard data sets drawn from real-world examples as well as on the data set used by NETtalk - the famous connectionist text-to-speech conversion system (Shavlik, Mooney, and Towell, 1991).

All data sets required the use of vector representations to encode the information. One data set contains 289 examples of 17 different soybean diseases. Each disease condition is

described by a distributed representation with 50 microfeatures such as weather, time of year, descriptions of leaves and stems, etc. Another data set contains 591 chess endgame examples belonging to two classes - win and not win - described with 36 features. The audiology data set consists of actual clinical cases - 226 examples of 24 categories of hearing disorders described with 58 features. This data set also had a large amount of missing information. The heart disease data set was also drawn from actual clinical examples. It had 303 examples belonging to two classes, described by 14 features. Six of these features (attributes) were numerically valued. And finally, the NETtalk training set, slightly modified, consisted of 4259 examples of parts of words classified into 115 phoneme/stress categories or outputs. (A subset of this data set - NETtalk-A - involving only the A sounds had 444 examples falling into 18 categories.)

ID3, a multilayered ANN using Backpropagation as well as a one-layered perceptron, were first trained on a subset of each data set and then were tested on their predictions of the unseen examples drawn from each data sets. The results of the experiment are reported in Table 2.4. (See Table 2.4.)

As can be seen from the results of this experiment the accuracy of predictions on novel examples of both ANN running BP and ID3 is almost the same. In some cases BP performed better (soybeans, heart disease and NETtalk-A) while in some others (chess and NETtalk-full!) ID3 was superior. In only two

Table 2.4

Experimental Comparison of ID3 and an ANN Using BP

Data set	Accuracy on test data		Difference in accuracy statistically significant (t-test)	ANN training time as a multiple of ID3 training time
	ID3	ANN using BP		
Soybeans	89%	94.1%	Yes	50
Chess	97%	96.3%	No	1000
Audiology	75.5%	77.7%	No	200
Heart disease	71.2%	80.6%	Yes	500
NETtalk-A	63.1%	66.4´	No	100
NETtalk-full	64.8%	63%	No	5

Note. Adapted from "Symbolic and Neural Learning Algorithms: An Experimental Comparison" by J. Shavlik, R. Mooney, and G. Towell, 1991, Machine Learning, vol. 6, pp.111-143.

cases are the differences in accuracy statistically significant.

But it is important to note that the relative training times of ID3 were much better - its average training time was 150 times shorter than the training time required for BP! This difference between the two algorithms is unlikely to be affected by using parallel hardware. ID3 is a recursive divide-and-conquer algorithm which can be implemented in parallel with significant gains in speed (cf. Shavlik, Mooney, and Towell, 1991, p. 136).

Similar experimental comparisons have been carried out by Fisher and McKusick (1989). They confirm that despite the great differences ANNs using BP and ID3 are able to solve the problems with only small differences in their degree of accuracy but with great differences in speed and in the transparency of the results.

Other symbolic induction systems have been shown to be very successful in processing large scale distributed representations which Smolensky would consider as belonging to the domain of subsymbolic computation. Atlas et al. (1990) compared the performance of a multilayered ANN using BP with the symbolic classification and regression tree system CART (Breimen et al., 1984). The real-world problems on which the two systems were tested included the prediction of the power consumption load for the Puget Sound Power and Light Company and predicting the power system security. Another task was

speaker independent vowel classification. All involved distributed representations with a great number of attributes or microfeatures. Table 2.5 presents the results of three experiments. (See Table 2.5.)

Similar comparative experiments between ANNs using BP and CART and ASSISTANT have been carried out by Weiss and Kapouleas (1989). They found that ANN using BP performed marginally better on the iris and the appendicitis data sets mentioned earlier but CART and ASSISTANT outperformed ANN using BP on the cancer data set (Weiss and Kapouleas, 1989, pp.784-85).

These comparative results explode the myth about the advantages of subsymbolic systems over symbolic systems. It turns out that ANNs are not necessarily softer, or more flexible than symbolic systems, and that they do not handle noisy data any better than symbolic systems. They show quite convincingly that the ability to use (numerical) vector representations (i.e. distributed representations and microfeatures) is not unique to ANNs. In this sense both ANNs and symbolic systems are subsymbolic. This term, however, is misleading because the "subsymbolic computation" carried out by systems like ID3 is entirely symbolic.

Despite the fact that systems based on TDIDT and BP can achieve very similar levels of success, they are by no means equivalent. The differences in their error rates on the different data sets reveal that they have different inductive

Table 2.5

Experimental Comparison of

CART and an ANN using BP

Problem	Error rates		Statistically significant
	ANN	CART	
Power consumption forecasting	1.39%	2.86%	No
Power system security prediction	0.78%	1.46%	Yes
Speaker-independent vowel classification	52.6%	53.6%	No

Note. Adapted from "Performance Comparisons Between Backpropagation Networks and Classification Trees on Three Real-World Applications" by L. Atlas, R. Cole, J. Connor, M. El-Sharkawi, R. Marks, Y. Muthasamy, E. Barnard, 1990, in D. Touretzky (Ed.), Advances in Neural Information Processing Systems, Vol. 2, San Mateo, CA: Morgan Kaufmann Inc., pp. 622-29.

biases and different generalization capacities, i.e. some problems "naturally" fit the inductive bias of one or the other algorithm. It is not difficult to create artificial data sets that better fit the inductive bias of ANNs using BP or of ID3 (Fisher and McKusick, 1989). Such inductive biases, however, do not appear to follow the hard/ soft distinction that Smolensky thinks underlies the difference between symbolic and subsymbolic systems. If symbolic systems like ID3 were naturally hard systems and if subsymbolic systems were naturally soft systems one would expect to see BP perform better than ID3 on the allegedly soft NETtalk speech generation task. Actually, the experiments show that the opposite is true, underlining the fact that there is no systematic difference in the inductive biases and in the generalization capacities of ID3 and the ANNs along the hard/ soft distinction.

The important difference between symbolic decision tree systems and ANNs using BP lies not in the symbolic/subsymbolic distinction but in the complete opaqueness of the way ANNs represent acquired knowledge. Whereas a decision tree can readily be converted into production rules -- a very versatile knowledge representation format -- no such possibility exists in the case with ANNs using BP. In fact, the weight matrices and the activations of the "hidden" units in which the knowledge acquired by ANNs is stored are usually viewed as a black box. A potential advantage of induction systems like ID3

over BP is that they represent explicitly the knowledge that they have acquired.

Conclusion

I think I have shown that Smolensky's symbolic/ subsymbolic distinction is entirely spurious. The differences between the inductive biases of the ANNs using BP and of other symbolic algorithms (most notably TDIDT systems) are not systematic in any respect. They cannot therefore, warrant the drawing of a principled, theoretically useful distinction between subsymbolic, subconceptual, soft and symbolic, or hard systems. In particular I have shown that

A1 is false because there are symbolic learning systems which are not brittle and which can be very flexible. They can model true human expertise in many areas (e.g. medicine, agriculture, banking, chess playing, water management - to list just a few).

A2 is false because the process of articulating expert knowledge in rules does not seem impractica. for many important domains (e.g. commonsense). In some cases it is possible to articulate commonsense knowledge in rules of thumb derived from pruned decision trees.

A3 is incorrect and B3 is false because it reflects the incorrect assumption that connectionist systems have contributed essentially more insight into how knowledge is represented in the brain than symbolic systems. A3 has to be rewritten as A3*.

B1 is false. Subsymbolic systems are not unique in their ability to represent concepts as sets of subconcepts or microfeatures. Ordinary symbolic systems can encode concepts as numerical state vectors.

B2 is false. Ordinary symbolic systems in many cases fare much better than subsymbolic systems in handling noisy, incomplete data and in displaying graceful degradation of performance in the presence of noise or incomplete information. The differences in the inductive biases of ANNs using BP and TDIDT symbolic systems are not systematic; these differences do not support the soft/ hard, symbolic/ subsymbolic distinctions.

CHAPTER THREE

Answering the Eliminativist Challenge: The Importance of Explicit Representations

Introduction

In Chapter One, we saw that the supporters of the classical approach to cognition (Fodor and Pylyshyn, 1988; Pylyshyn, 1984) and implementational connectionists (Smolensky, 1991; Touretzky, 1991) disagree on two main fronts -- the extent to which neural networks can implement symbolic structures, and the significance of connectionism for cognitive psychology should it turn out that neural networks can fully implement the classical Turing/ von Neumann cognitive architecture. Despite their disagreements, implementational connectionists and the supporters of the classical approach agree on one major issue; they believe that no cognitive theory can ever hope to be successful unless it is able to explain such pervasive cognitive phenomena as systematicity. So, implementational connectionists do not doubt the need to implement in neural networks such symbolic structures as production rules, parse trees, part-whole hierarchies, etc. that would enable neural networks to process complex symbolic representations.

Implementational connectionism, however, is by no means recognized as the 'orthodox' connectionist cognitive theory. The doubts expressed by Fodor and Pylyshyn regarding its

psychological significance are shared by the great majority of connectionists researchers who do not believe that symbolic structures of any kind play a role in cognition. The majority of the connectionists who see artificial neural networks as providing the basis for explaining human cognitive architecture subscribe to the theory that in a mature cognitive science there will be no room for any of the symbolic structures postulated by present-day cognitive science -- human behaviour will be explained solely in terms of patterns of activation and weight changes in neural nets. This theory, aptly named eliminative connectionism (by Pinker and Prince, 1988), is a much more radical doctrine than implementational connectionism. If true, it should force a major reconceptualization of our current psychological theories.

Eliminativism

Eliminative connectionism is a species of a more general view -- eliminativism. Eliminativism is the claim that

> some category of entities, processes or properties exploited in a commonsense or scientific account of the world do not exist (Ramsey, Stich, and Garon, 1991, p. 201).

Eliminativism with respect to psychology is the claim that our commonsense notions (of belief, desire, expectation, fear, etc.) that feature prominently in our 'folk' psychological

theory of ourselves simply do not refer to anything. This type of eliminativism with respect to folk psychology is common to behaviourism (Skinner, 1953, 1957), eliminative materialism (Churchland, 1988, 1989), and to the syntactic theory of the mind pioneered by Stich (1983). All of these different theories of the mind are opposed to any attempt to treat beliefs, desires or any other of the so called propositional attitudes as legitimate candidates for possible scientific reduction. Accordingly, eliminativists of different schools reject both central state identity materialism (Armstrong, 1968) as well as the functionalist form of materialism (Putnam, 1960; Fodor, 1975; Pylyshyn, 1984) that hopes to reduce the beliefs and desires of folk psychology to physio-chemical or computational states of the brain.

Eliminativism (especially in psychology) relies on one basic type of argument that is entirely analogical. History of science has shown us -- the argument goes -- that certain entities like caloric, phlogiston, witches, demonic possession, etc. postulated by successful scientific or folk theories, have turned out to be entirely spurious. The case with the folk psychology is analogous. As later scientific theories showed that there are no such things as caloric and phlogiston, future behaviourial, neurophysiological, or computational theories will show that there are no such things as beliefs and desires. This patently analogical argument is usually backed-up with the assertion that folk psychology

cannot be considered true because there are abundantly many undisputable cases in which folk psychological explanation either fails or is vacuous.

But is this argument sufficient to show that beliefs and desires are spurious entities? It is well known that a theory may be false or it may fail to explain certain cases even though its central terms have reference; a theory change need not always imply the spuriousness of the entities postulated by the old theory. Although Lavoisier's oxygen theory showed the spuriousness of phlogiston and kinetic energy of modern thermodynamics replaced caloric. But molecular genetics did not replace the genes postulated by the older genetic theory. History of science demonstrates that radical theory shifts co-exist with more gradual, reductionist theory changes in which the central terms of the old theory are not replaced but are made more precise by the new successor theory.[3] What is the indication that folk psychology will be eliminated by its successor theories rather than gradually made more precise as was the case with molecular genetics? There is <u>nothing</u> the eliminativist can point to in order to demonstrate that the fate awaiting folk psychology is total elimination rather than gradual reduction -- the historical <u>analogies</u> supporting elimination are just as numerous as the <u>disanalogies</u>. But if that is so, eliminativism <u>per se</u> is simply a statement about what the future of folk psychology <u>might</u> turn out to be: eliminativism has no evidence on its side to allow us to claim

that elimination must be the future of folk psychology. The advent of connectionism changed all that. Connectionism provides a hope that much needed additional evidence might be closer than critics and defenders of eliminativism had previously thought.

Eliminative Connectionism

Eliminative connectionism is a psychological theory that denies that the entities referred to by the traditional cognitive psychology -- symbolic structures such as rules, parse trees, propositions, etc. -- actually exist. Eliminative connectionism has direct implications for eliminativism with respect to commonsense psychology. This is so because traditional cognitive psychology and folk psychology share a common basis which Stich (1983) has termed the propositional modularity view of the mind. For Stich this view amounts to the claim that

> propositional attitudes are functionally discrete, semantically interpretable, states that play a causal role in the production of other propositional attitudes, and ultimately in the production of behaviour. (Ramsey, Stich, and Garon, 1991, p.204.)

Eliminative connectionism denies that there are any such functionally discrete representational states that are even remotely identifiable with the propositional attitudes of traditional cognitive psychology. In contrast to the old

eliminativism, however, the reasons given in defence of eliminative connectionism are not historical analogies; they are derived directly from the architectural differences between connectionist and symbol processing cognitive models. This would seem to give much needed support to the eliminativist program in general: if for some nontrivial cognitive tasks there exist connectionist models that do not implement explicit rules and symbolic representations and at the same time there exist no corresponding symbolic models that can solve the same task, then with the accumulation of such evidence we will have good reason to accept the eliminativist view that rules and symbolic representations have no place in cognition. And if eliminative connectionism turns out to be right, so too will eliminativism about propositional attitudes (cf. Ramsey, Stich, and Garon, 1991, p.200).

There are three connectionist claims that are widely assumed to bear directly on the eliminativist thesis:

T1. Connectionist representations are not functionally discrete.

T2. Connectionist representations are not (directly) semantically interpretable.

T3. ANNs are primarily pattern associators and pattern recognizers.

The first claim is based on evidence from those

connectionist models that employ widely distributed representations. The crucial assumption here is that ANNs can be seen to represent information in their connection weights and in the biases of their units in distributed form, i.e. in such a way that human interpreters can tell what information the network is representing as a whole, while they are incapable of identifying which weights and biases represent specific parts of this information. This implies that any attempt to isolate a set of weights and biases as encoding a given proposition or set of propositions is bound to fail in the case of genuinely distributed representations. But if the distributed nature of connectionist representations implies that they have no functionally separable individual states, then propositional attitudes can have no causal role. As Ramsey, Stich, and Garon put it

> It simply makes no sense to ask whether or not the representation of a particular proposition plays a causal role in the network's computation. (Ramsey, Stich, and Garon, 1991, p.212.)

Second, many connectionists (most notably Smolensky, 1988) believe that individual units in multilayer ANNs which have no obvious symbolic interpretation can be considered as encoding subsymbolic representations. The whole pattern of activation of neural networks can be given a symbolic interpretation which will be roughly right but imprecise --

the fine structure of cognition, the structure that is causally relevant is to be found at the subsymbolic level. Subsymbolic representations are descriptions built up of entities that correspond to constituents of symbols; these fine-grained constituents are called subsymbols, and they are the activities of individual processing units in connectionist networks. (Smolensky, 1988, p.3). The interaction between these activity patterns is not directly described by a formal definition and can be computed only approximately.

The third claim is based on the fact that for many purposes ANNs can be viewed as pattern associators and pattern recognizers. But if ANNs can solve typical pattern association or pattern recognition tasks without explicitly encoding rules and symbolic representations, then this is prima facie evidence that symbol structures are redundant from a connectionist perspective.

T1, T2, and T3 imply that distributed representations in ANNs which act as pattern associators and pattern recognizers are incompatible with symbolic structures like propositions, rules, parse trees, etc., So, if human cognitive architecture were based entirely on neural networks with distributed representations and if the operations of pattern association and pattern recognition were the basic operations of human cognitive architecture, then it would seem that we have to rule out the propositional attitudes of commonsense psychology as non-entities.

Let's suppose that the reasoning so far is correct. We have established the conditional claim that if eliminative connectionism turns out to be right, so too will eliminativism about propositional attitudes. But we still need to know under what conditions connectionism might turn out to be right. Eliminative connectionism will succeed as a theory of the mind only if it can be demonstrated that human cognitive architecture is in fact connectionist and that the correct connectionist architecture does not, in fact, implement any symbolic structures. Only if these two conditions are met can we conclude that commonsense psychology is a radically false doctrine and that its central terms are non-referential. But in order to have a complete argument for the elimination of folk psychology we need to know whether connectionist ANNs that do not implement symbolic structures can serve as a basis for the human cognitive architecture.

As things stand at the moment, the eliminative connectionists can only point to several successful connectionist models which they claim make no explicit use of symbolic structures like rules and propositions -- models for which ANNs are used as pattern associators and pattern recognizers. These eliminative connectionist models can be divided roughly into two groups -- illustrative models that are intended to serve as mere visual aids in illustrating how neural networks can solve problems without implementing any symbolic structures, and more or less serious working models

that try to solve nontrivial cognitive problems without implementing symbolic structures. In the first group are many of the models discussed in the philosophical literature (cf. Churchland, 1989; Ramsey, Stich, and Garon, 1991).

A good illustrative example of how networks can avoid symbol structures like propositions and still solve cognitive tasks has been given by Churchland (1989). He has speculated that cognition can best be explained in terms of prototype activation. He sees in the current connectionist research the kernel of a new conception of cognitive activity,

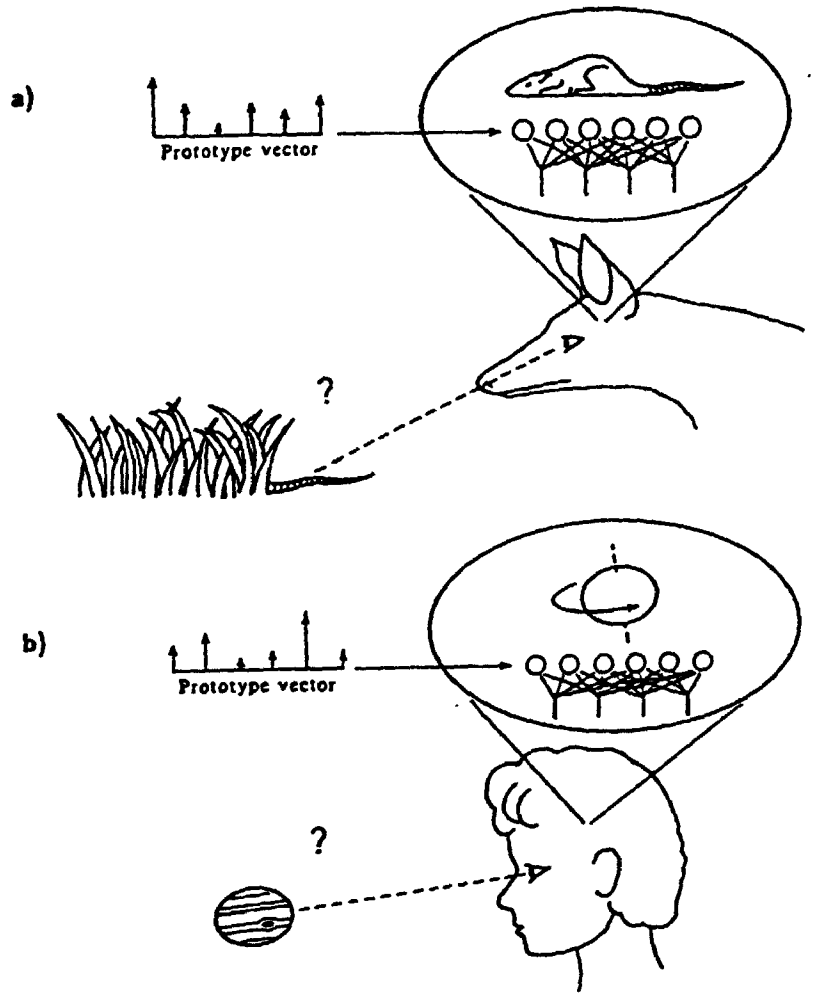
> ...a conception in which vector coding and vector-to-vector transformation constitute the basic forms of representation and computation, rather than sentential structures and inferences made according to structure sensitive rules. (Churchland, 1989, p.209.)

According to this conception pattern recognition is the basic operation in cognition that underlies human and animal explanatory understanding. Things as different as desert rats and rotating plastic objects are understood by human and animal minds when their perceptual systems activate the prototype vectors encoding desert rat and rotating plastic object. (See Figure 3.1.) In Churchland's opinion

> Explanatory understanding consists in the activation of a specific prototype vector in a well-trained network. It consists in the apprehension of the problematic case as an instance of a general type, a type for which the

Figure 3.1

Explanatory understanding as the activation of a prototype vector. a) Ampliative activation of desert rat vector. b) Ampliative activation of rotating plastic body vector.


a)
Prototype vector
?
b)
Prototype vector
?

creature has a detailed and a well-informed representation. (Churchland, 1989, p.210.)

Unfortunately, Churchland gives no estimate of the number of possible general prototypes and he does not say how they relate to each other or to the vectors that activate them; he does not offer any similarity metric or any other gauge with which to measure when a given pattern does or does not belong to a given general prototype. This, however is not a mere detail that could be sorted out later, without an indication of how an ANN can organise all the patterns it is recognizing in a systematic way Churchland's vision of cognition as pattern association is nothing more than an unsubstantiated claim.

Ramsey, Stich, and Garon have given another example of how the possible elimination of symbolic structures might occur in connectionist ANNs. Their model involves learning and recognizing the truth or falsity of 16 propositions (see Table 3.1.)

Ramsey, Stich, and Garon trained a three layer ANN to associate each input pattern with its corresponding value 1 or 0 (true or false). The network was then able to predict the truth value of proposition #17 which was not in the training set. Ramsey, Stich, and Garon point out that this network cannot be regarded as even implicitly encoding such things as propositions and rules. They compare their network with a

Table 3.1

Propositions and Truth-Values

	Proposition	Input	Output
1	Dogs have fur.	1100001100001111	1 true
2	Dogs have paws.	1100001100110011	1 true
3	Dogs have fleas.	1100001100111111	1 true
4	Dogs have legs.	1100001100111100	1 true
5	Cats have fur.	1100110000001111	1 true
6	Cats have paws.	1100110000110011	1 true
7	Cats have fleas.	1100110000111111	1 true
8	Fish have scales.	1111000000110000	1 true
9	Fish have fins.	1111000000001100	1 true
10	Fish have gills.	1111000000000011	1 true
11	Cats have gills.	1100110000000011	0 false
12	Fish have legs.	1111000000111100	0 false
13	Fish have fleas.	1111000000111111	0 false
14	Dogs have scales.	1100001100110000	0 false
15	Dogs have fins.	1100001100001100	0 false
16	Cats have fins.	1100110000001100	0 false
Added Proposition:			
17	Fish have legs.	1111000011001000	1 true

Note: Adapted from "Connectionism, Eliminativism, and the Future of Folk Psychology" by W. Ramsey, S. Stich, and J. Gasron, 1991, Ramsey et al. (Eds.) Philosophy and Connectionist Theory, pp.199-228, Hillsdale, NJ: Erlbaum.

traditional semantic network that uses such symbolic structures as propositions and inheritance relations to demonstrate how dissimilar they are even though their performance on this limited classification task is similar. According to Ramsey, Stich, and Garon this is a reason to believe that such connectionist networks that work without any symbolic structures might turn out to be better models of memory and judgement than inheritance systems which are committed to using functionally discrete propositions.

The question that hangs over such examples of elimination of rules and symbolic representations, however, is how realistic it is to expect that these "toy" models will scale up so that they will be able to account for serious nontrivial cognitive tasks. Neither Churchland nor Stich address seriously this question. But this is by no means a secondary issue. It is not enough just to point out that the brain could possibly encode $_{10}10^{8}$ different vectors (Churchland, 1989) without any indication how this enormous number of encodings can be organized. We need to see massive evidence that ANNs can solve nontrivial cognitive tasks and that they can integrate and further process the results of these solutions. The models suggested by Churchland and by Ramsey, Stich, and Garon fall far short of any such serious scrutiny; they are offered not as serious cognitive models but for purely illustrative purposes. Such models do not advance the eliminative cause any further than the historical and

analogical arguments employed previously. They serve as examples of what would be the case, were it to turn out that human cognitive architecture is connectionist.

There are eliminative connectionist models however, that try to provide successful connectionist accounts of nontrivial cognitive tasks. One of the most interesting and controversial areas where such connectionist models have been proposed is the area of language learning and language processing. I will examine two of the most influential models in this area of connectionist research -- Rumelhart and McClelland's and MacWhinney and Leinbach's models of learning the past tenses of English verbs. I will demonstrate that those models which try to solve nontrivial cognitive tasks are seriously flawed and so cannot be taken as evidence that ANNs as pattern associators and pattern recognizers can solve nontrivial cognitive tasks. Ultimately, the analysis of Rumelhart and McClelland's and MacWhinney and Leinbach's models should give us sufficient reason to reject them as examples of successful elimination of explicit rules and symbolic representations from cognitive science.

An Examination of the Eliminative Connectionist Conception of Language Learning and Language Processing

Ever since the publication of N. Chomsky's Syntactic Structures (Chomsky, 1957) and his subsequent attack on a behaviourist theory of language (Chomsky, 1959), it has

generally been assumed that rules and symbolic representations like parse trees are indispensable for a successful account of language learning and language processing. But according to the eliminative connectionists (Rumelhart and McClelland, 1986; McClelland, Rumelhart and Hinton, 1986) it is impossible to find a principled mapping between the connection strength matrixes and vectors of activations used in a connectionist ANN, and symbol structures like parse trees, propositions, and the rules for their manipulations which one finds in symbolic models of language processing. At the same time, the supporters of this approach believe that the resources of connectionist architectures are sufficient to explain in principle all psycho-linguistic phenomena without the need to postulate the existence of explicit rules and symbolic representations.

Eliminative connectionism has received a great deal of support from connectionist research on language that directly challenges the rules and representations accounts of language learning and language processing. In particular, the task of learning the past tenses of English verbs has received a great deal of attention. The first PDP model developed by Rumelhart and McClelland was, according to its creators, able to learn the past tenses of English verbs without the use of explicit rules and symbolic representations (Rumelhart and McClelland, 1986, pp. 216-271). After several critical reviews of this model by Pinker and Prince (1988), Lachter and Bever (1988)

and a detailed background analysis of the model by Plunkett and Marchman (1991), a new revised PDP model was proposed by MacWhinney and Leinbach (1991). MacWhinney and Leinbach claimed that their new model meets most of the criticisms addressed at the earlier eliminativist model of Rumelhart and McClelland and that it is also able to learn the past tenses of English verbs without any explicit representation of the acquired linguistic knowledge. Rumelhart and McClelland as well as MacWhinney and Leinbach argued that since there was no comparable symbolic model that can achieve similar results by using explicit rules and symbolic representations and since it was difficult to imagine how 'rigid' rules could ever account for the flexibility of human language learning, these PDP models should be seen as crucial evidence in favour of eliminative connectionism.

A common feature of both Rumelhart and McClelland's and MacWhinney and Leinbach's models is the treatment of the ANN as a device that can learn to associate arbitrary patterns. Both models make much use of the fact that a multilayer ANN using the error-backpropagation algorithm or a perceptron using the perceptron convergence algorithm can be "conditioned" to associate pairs of arbitrary patterns. In this process, which is essentially a supervised learning from examples, pairs of input and output patterns are provided by the human supervisor or "teacher". A properly structured multilayered ANN can be trained to associate each input

pattern with each output pattern using the error backpropagation algorithm in the following way:

> The ANN (with usually randomized weights) first uses the input vector (pattern) which is propagated forward to produce its own output vector (pattern). Then it compares its output with the target vector (this is the pattern it has to learn). If the differences between the output vector and the target vector are similar within a specified range the training is stop$_t$ed. If the differences are outside of this range, the difference is reported as an error signal for each output unit. Then the error signal is passed backwards to each unit in the network and the appropriate weight-changes are made. The forward and backward procedures form one epoch. Such epochs are repeated until the differences between the input and the target vector are minimized to a desired level and the human controller terminates the process. (The perceptron convergence procedure for single layer ANNs is very similar.) (Cf. Rumelhart, Hinton, and Williams, 1986, p. 327.)

After an ANN has been trained, if we present it with any of the input patterns, it can (in many cases) produce the correct or desired output pattern. Then we can say that the network has learned to associate the two sets of input and output patterns. Moreover, networks trained in such a way

appear able to extract (some) regularities that exist in the pairings of input and output patterns and sometimes can respond correctly when presented with input patterns not encountered during training. This suggests that (in some cases) they exhibit a degree of genuine inductive generalization abilities.

According to eliminative connectionism the significance of the ANN as a pattern associator for psychology lies in the fact that on the one hand ANNs can learn to correlate arbitrary patterns and to generalize their knowledge to other "unseen" patterns, but on the other hand, they cannot and need not represent the acquired knowledge in symbolic form; they cannot and need not consult any explicit rules during or after the acquisition of this knowledge. In fact, the "representation" of knowledge in terms of connection strengths and patterns of activation is in purely numeric form and is entirely opaque to the human interpreter. In this way the ANN as a pattern associator solves the well known problem of knowledge representation by simply dissolving it. Thus, one of the main claims of eliminative connectionism is that, with the development of powerful connectionist models, the whole of cognition (including language learning and language processing) will eventually come to be seen simply as process of pattern association and pattern recognition. In this way all reference to rules and symbolic representations will be eliminated from cognitive science.

In what follows I will show that these connectionist models do not offer evidence sufficient to establish the claims of eliminative connectionism and a fortiori, they do not establish the claims of general eliminativism either.

Rumelhart and McClelland's Model

Rumelhart and McClelland provided the first step in what they see as the gradual elimination of the use of rules and symbolic representations in cognitive psychology by developing a PDP model that was credited by them with the ability to learn the past tenses of English verbs. They specifically addressed the issue of the existence of explicit but inaccessible rules in language acquisition associated with the work of N. Chomsky, S. Pinker, and other supporters of the symbol-processing approach. According to Rumelhart and McClelland, their PDP model eliminates the need to postulate the existence of explicit rules in accounting for the human knowledge of language:

> We suggest instead that the implicit knowledge of language may be stored in connections among simple processing units organized into networks. While the behaviour of such networks may be describable (at least approximately) as conforming to some system of rules, we suggest that an account of the fine structure of the phenomena of language use and language acquisition can best be formulated in models that make reference to the

characteristics of the underlying networks. (Rumelhart and McClelland, 1987, p. 196.)

I will not describe in any great detail Rumelhart and McClelland model since this model is one of the best known connectionist simulations. I will only note that at the heart of their model is a simple perceptron-based pattern associator. This pattern associator is connected to an input encoding network and an output decoding/ binding network. (See Figure 3.2.)

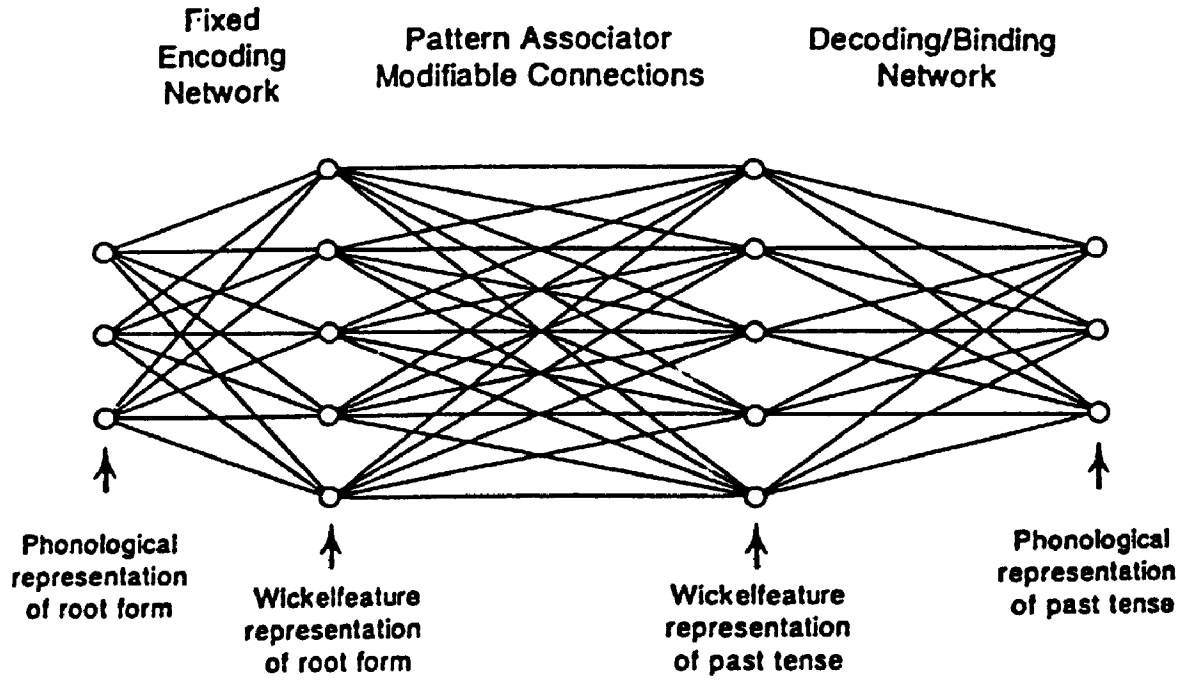
The encoding network takes as input the phonological representation of the root form of the verbs and converts it into a special Wickelfeature representation format. The Wickelfeature representation of the root form of each verb is then paired with the Wickelfeature representation of its past tense in the perceptron pattern associator for the duration of the training process. The decoding/ binding network is used to decode the output of the pattern associator from the Wickelfeature format back into the same phonological format used at the input of the encoding network.

The Wickelfeature representational format played an important role in Rumelhart and McClelland's simulation. It derives from a scheme proposed by Wickelgren (1969) that represents each phone in a word as a triple, called a Wickelphone for short, consisting of the phone itself, its predecessor, and its successor. A phoneme occurring at a word

Figure 3.2

Rumelhart and McClelland's network.

(Reprinted from Rumelhart and McClelland, 1986.


Fixed
Encoding
Network
Pattern Associator
Modifiable Connections
Decoding/Binding
Network
Phonological
representation
of root form
Wickelfeature
representation
of root form
Wickelfeature
representation
of past tense
Phonological
representation
of past tense

boundary has a special boundary symbol (#). So the word cat /kat/ can be represented as the set of Wickelphones (#ka, kat, at#). However, the direct use of Wickelphones in this simulation was computationally unfeasible because the number of possible Wickelphones required the number of connections in the network to be on the order of 10^9! That is why Rumelhart and McClelland decided to represent each phoneme not by a single Wickelphone but by a pattern of what they called Wickelfeatures. Essentially, this format classifies all phonemes into different types, e.g. interrupted consonants, continuous consonants, vowels, etc. that are further subdivided into stops and nasals or fricatives, liquids, etc. Using this classification each phoneme can be represented by a unique combination of category features, i.e. as an 11-bit binary vector (including the boundary marker). Then each Wickelphone can be represented as a 33-bit binary vector. However, in order to make the decoding of Wickelphones manageable Rumelhart and McClelland selected only 460 of the 1210 possible Wickelfeatures for the actual experiment. One particular drawback of this decoding decision was that sometimes two or more words had to "compete" for one and the same Wickelfeature representation, i.e. the network produced two or more past tense forms for one and the same stem. In such cases Rumelhart and McClelland took the response strength of each of the alternatives as the indication of the likelihood that the model would actually output the correct

candidate. If the strength of a response was weaker than 0.2 (on 0 to 1 scale) it was counted as no response.

Rumelhart and McClelland claimed that after two limited training periods the network was able to learn correctly the past tenses of both the regular and irregular verbs used for training and to generalize this knowledge to previously "unseen" regular and irregular verbs. Moreover, the errors made by the model during the training process broadly followed the U-shaped learning curve in the acquisition of the past tense exhibited by young children: early -- correct production of regulars and irregulars, medium -- incorrectly regularized irregulars, and late -- correct production of the majority of regulars and irregulars.

According to Rumelhart and McClelland the success of their model confirms one of their main predictions -- the nonexistence of explicit rules and symbolic representations mediating language acquisition:

> We have, we believe, provided a distinct alternative to the view that children learn the rules of English past-tense formation in any explicit sense. We have shown that a reasonable account of the acquisition of past tense can be provided without recourse to the notion of a "rule" as anything more than a description of the language. We have shown that, for this case, there is no induction problem. The child need not figure out what the rules are, nor even that there are rules. The child need not

> decide whether a verb is regular or irregular. There is no question as to whether the inflected form should be stored directly in the lexicon or derived from more general principles. (Rumelhart and McClelland, 1986, p. 267.)

Criticism of Rumelhart and McClelland's Model

Rumelhart and McClelland's model has received extensive critical attention in the literature. Two of the best known critical reviews are Pinker and Prince (1988) and Lachter and Bever (1988). The most important conceptual issues to surface during the discussion were connected with the support that the eventual success of Rumelhart and McClelland's model of learning the past tenses of English verbs (or similar improved PDP models) could lend to eliminative connectionism.

The induction problem. Rumelhart and McClelland believe that one of the major results of their simulation is the demonstration that in the learning of the past tenses of English verbs there is no induction problem. But even if their model had a 0% error rate, such a conclusion is absolutely unjustifiable. As Pinker and Prince, and Lachter and Bever point out, Rumelhart and McClelland try to model the acquisition of the production of the past tense considered in isolation from the rest of the English morphological system. Rumelhart and McClelland assume that the acquisition process establishes a direct mapping from the phonetic representation

of the stem to the phonetic representation of the past tense form (cf. Pinker and Prince, p. 87-88). This direct mapping collapses some well established distinctions:

- Lexical item vs. phoneme string. The lexical item combines syntactic, semantic, morphological and phonological properties whereas the phoneme string encodes just one of these properties. As the existence of homophones with different past tense forms (e.g. wring/ wrung vs. ring/ rang) indicate, phonetic input does not entirely control lexical access and there is no complete overlap between purely phonetic representation and lexical representation; there is still an inductive step to be made from phonetic information to lexical information.
- Morphological category vs. morpheme. There is a huge inductive step to be made in passing from a simple morpheme to its morphological category such as 'past tense', 'present tense', etc. (Pinker and Prince, p. 86). This inductive step can remain hidden from us if we are not aware of the conditions under which <u>supervised</u> learning from examples takes place. The training process in this learning task assumes a supervisor who "knows" the correct classifications to all training examples, i.e. the supervisor has direct access to category information. It is only if one forgets about the role of the supervisor in the training process that one can say

> that for ANNs using backpropagation there is no induction problem. It should be quite clear that the system does not end up inferring the category of past tense from the training sample. It only learns to associate phonetic patterns and does not make the inductive step to new morphological categories.

So even if the model was fully successful in achieving the desired direct mapping from input phonetic strings to output phonetic strings, that would not mean that there is no induction problem in the learning of the past tenses of English verbs. At least two crucial steps are still required to go from the phonetic strings to lexical items and then to morphological categories like 'present tense' and 'past tense'. It is unclear, however, how a PDP model that acts only as a pattern associator will be able to learn new category information. Simply remaining at the level of phonetic patterns, it is impossible to express the new categorial information necessary for further inductive steps. As Pinker and Prince put it:

> [one of the inherent deficits of the model is that] there is no such thing as a variable for any stem, regardless of its phonetic composition, and hence no way for the model to attain the knowledge that you could add /d/ to a "stem" to get its past. Rather, all the knowledge of the model consists of responses trained to the concrete

> features in the training set. (Pinker and Prince, 1988, p. 124.)

This criticism pinpoints the upper limitations of Rumelhart and McClelland's model -- it is not able to acquire knowledge that can be represented in first-order (function/ variable) format. The model lacks the capacity to express simple generalizations like this one:

> For any verb stem, if it ends at /r/ or /l/, add /d/ to form its past tense; if it ends at /p/ or /k/, add /t/ to form its past tense.

Representational format. The representational format used by Rumelhart and McClelland in their model was also subjected to severe criticism. Lachter and Bever pointed out that the results the model actually achieved would have been impossible without the use of several TRICS (The Representations It Crucially Supposes). One such TRIC was reducing the number of Wickelfeatures from about 1000 to 260 not randomly but in such a way as to decrease disproportionally the information provided by some Wickelfeatures. This selective deletion had the effect of reducing the information contained in some Wickelphones and increasing it in the "centrally informative" Wickelphones (Lachter and Bever, 1988, p. 209). This certainly introduced a bias in the data favourable to the success of the model. Also, explicitly coding word boundary information in a completely separate set of 200 Wickelfeatures had the effect

of giving privileged information status to phonemes in the beginning and at the end of a word. This "boundary sharpening" also introduced a favourable bias (Lachter and Bever, 1988, p. 210). There are further questions about the linguistic and psychological justification for using this specific encoding of Wickelphones in phonetic features and about the use of the Wickelphone representational format itself. The probabilistic decoding of two or more candidates for an answer also seems to be a puzzling feature of the model that is without any psychological raison d' etre.

Pinker and Prince note other problems associated with the representational format chosen by Rumelhart and McClelland. Wickelphone representations do not always preserve natural sequential order. For example the word strip can be represented either as (#st, str, tri, rip, ip#) or as the unordered (ip#, rip, str, #st, tri) (Pinker and Prince, p. 89). Also, this format cannot encode all possible words unambiguously. For example, the model would be unable to distinguish between the words algal and algalgal in the Australian language Oykangand. This means that with this format we cannot represent strings of arbitrary lengths.

Performance Results

On purely conceptual grounds I am not convinced that Rumelhart and McClelland have provided an adequate model of the learning of the past tenses of English verbs because of

the inductive limitations built in their model. But when one looks at the actual performance of the model one is surprised by Rumelhart and McClelland's claim that their model comes close to accomplishing the limited task of learning the set of specified mappings between phonetic patterns. Despite the many favourable biases intrcduced in the data with the use of TRICS, a close analysis of the reported experimental results shows that the model's performance is very poor.

Rumelhart and McClelland trained the network with 420 regular and irregular verbs that were specially chosen for the purpose, i.e. they did not use random sampling. The actual training of the network was carried out in two stages. First it was trained on only 10 high frequency verbs. In the second stage 410 medium frequency verbs were added. The psychological justification for this procedure is doubtful (cf. Plunkett and Marchman, 1991, p. 47). At the end of the training the network had learned approximately 98% of the regulars and about 95% of the irregulars (cf. Rumelhart and McClelland p. 246). The testing sample consisted of 86 "unseen" low frequency verbs (14 irregular and 72 regular) also not randomly chosen. The complete training process was extremely slow; it took 260 hours of computer time. The testing process of generating free responses from the network upon presentations of verb stems took 28 hours of computer time.

The results on the testing sample were: 93% error rate (!) for the irregulars, i.e. only one out of 14 irregular past

tense forms was correctly produced (e.g. the no change bid --> bid). For two verbs cling and weep, both a correct and an incorrect past tense candidate was produced. However, in both cases the incorrect candidates had the highest likelihood. One consolation for this very high error rate might be that most of the incorrect responses on the irregulars were regularization mistakes, like catch --> catched or were no change mistakes, like grind --> grind. The regulars fared better with a 33.3% error rate. Some of the mistakes there were of the kind that humans almost never make, like tour --> toureder, or mail --> membled. The overall errors for the whole testing sample was 37 wrong past tense forms out of 86 tested or an error rate of 43%[4]!

What is particularly puzzling is why for 6 regular verbs the network did not produce any past tense at all, i.e. all of the response strengths were under 0.2. As Pinker and Prince have noted "This suggests that the reason for the model's muteness is that it failed to learn the relevant transformations; i.e. to generalize appropriately about the regular past." (Pinker and Prince, 1988, p. 124). Even more difficult to explain is why a large number of mistakes on the regular verbs were not psychologically realistic but were mistakes that no human learner would make:

squat - squakt

mail - membled

tour - toureder

mate - maded

brown - brawned

shape - shipt

sip - sept.

Another puzzling error that is only rarely committed by humans was the doubling of past forms.

type - typeded

step - steppeded

snap - snappeded

drip - drippeded

carp - carpeded

smoke - smokeded.

Pinker and Prince believe that the fact that the trained model was producing such unusual or 'humanly impossible' errors "...implies that a major induction problem -- latching onto the productive patterns and bypassing the spurious ones -- is not being solved successfully" (Pinker and Prince, 1988, p. 125). I fully agree with such an assessment. The poor performance of Rumelhart and McClelland's model is not accidental; it stems from the fact that neural networks used as <u>pattern associators</u> cannot represent or cannot acquire knowledge in a function/ variable format and therefore are incapable of learning representations in first-order form.

Unfortunately, in the absence of a symbolic learning model that can match even the modest results achieved by Rumelhart and McClelland's connectionist model, eliminative

connectionists can dismiss important theoretical objections by typically claiming that "PDP models may provide more accurate accounts of human performance than models based on a set of rules representing human competence" (McClelland, Rumelhart, and Hinton, 1986, pp. 24-25). I will show that at least with respect to the learning of the past tenses of English verbs and in many other learning tasks, this need no longer be the case.

MacWhinney and Leinbach's Model

In a paper recently published in Cognition, MacWhinney and Leinbach (1991) report a new connectionist model of the learning of the past tenses of English verbs. They claim that the results from the new simulation are far superior to Rumelhart and McClelland's results and that they can answer all of Pinker and Prince's, as well as Lachter and Bever's, criticisms of the earlier model.

Input, Output Representation and the Structure of the Network

MacWhinney and Leinbach's major departure from Rumelhart and McClelland's model is the change of the representational format. The Wickelphone representational format is replaced with the UNIBET (MacWhinney, 1990) phoneme representational system which allows the assignment of a single-letter ASCII code to each phoneme. In this way the UNIBET format appears to avoid the context dependence of Wickelphones. However, as in Rumelhart and McClelland's model, the phonetic information

accessible to the network is to some extent context-dependent. Instead of coding predecessor and successor phonemes as Wickelphones MacWhinney and Leinbach use special templates with which to code each phoneme and its position. In their simulation all input verbs are represented in phonetic form with the help of a left justified and a right justified template of the form

CCCVVCCCVVCCCVVCCC VVCCC

left justified template right justified template

where C stands for consonant, and V for vowel space holders. For example, the verb bet is represented in UNIBET format as /bEt/ and using the template form as

bCCEVtCCVVCCCVVCCC VECCt

In this way all English verbs that have more than three syllables, more than three consonantal phonemes in a row, or more than two vocalic phonemes in a row are left out of this experiment because they fail to fit the chosen template.

Altogether 2062 regular and irregular English verbs were selected. Of these, 10% of the least frequently occurring regular verbs and 10% of the least frequently occurring irregular verbs, were set aside for testing the predictive success of the model, while the rest were used for training. There were 118 irregular past tense forms in the training sample; this included some of the most frequently occurring verbs.

The actual input to the network was created by coding the

individual phonemes as disjoint sets of phonetic features. There are 8 features for vowels and 10 for consonants.

Vowels:

i	front
I	centre, high
e	front, middle
E	front, middle, low
&	front, low
u	back, round, high
U	centre, back, round, high
o	back, round, middle
O	back, round, low
a	centre, low
6	centre, middle
1	front, back, round, high, low, diphthong
2	front, back, round, high, middle, diphthong
3	front, high, low, diphthong

Consonants:

p	labial
t	dental
k	velar
b	voiced, labial
d	voiced, dental
g	voiced, velar
m	voiced, labial, nasal
n	voiced, dental, nasal

N	voiced, palatal, velar, nasal
l	voiced, dental, palatal, liquid
r	voiced, dental, trill
f	labial, dental, fricative
v	voiced, labial, dental, fricative
s	dental, fricative
z	voiced, dental, fricative
S	palatal, fricative
Z	voiced, palatal, fricative
j	voiced, palatal, liquid, fricative
h	velar, fricative
w	labial, liquid, fricative
T	dental, fricative, interdental
D	voiced, dental, fricative, interdental

Each input and output unit stands for a particular feature and its activation indicates the presence of this feature, otherwise the feature is presumed absent. For example, the vowel U can be represented in the network as a pattern of activations in the following way:

-	+	+	+	-	-	+	-
front	centre	back	high	low	middle	round	diphthong

For the consonant Z we will need 10 units to represent the particular pattern of features corresponding to it:

+	-	-	+	-	-	-	-	+	-
voiced	labial	dental	palatal	velar	nasal	liquid	trill	fricative	interdental

In this way, for each vowel slot there are 8 dedicated units of the network and to each consonant slot there are 10 dedicated units, representing the values of different combinations of phonetic features. Altogether there are 214 feature/ slot input units plus 5 specially dedicated control units which code morphological category information that could be used to switch the response of the network to present, past tense, past participle, present participle, or third person singular. The output layer has only 168 feature/ slot output units, reflecting the fact that the output template does not have a right-justified part.

The network has two layers of 200 "hidden" units fully connected to adjacent layers. This number was arrived at through trial and error. As MacWhinney and Leinbach explain, "the model uses two layers of hidden units, because a model which had only one layer did not do as well at learning the training set" (MacWhinney and Leinbach, 1991, p. 143). In a departure from the standard practice, in similar experiments MacWhinney and Leinbach' s network had a special set of connections that allowed it to copy the left-justified phonological form of the input directly onto the output prior to learning. This feature allowed the creation of a bias in the output nodes that could facilitate the learning process, since in most cases present and past forms of English verbs differ slightly. The overall structure of the network is shown in Figure 3.3. (See Figure 3.3.)

The learning procedure used was the standard form of the error backpropagation algorithm. The network was trained to associate the phonetic patterns representing the verb stems with the corresponding correct form for the past tense.

Results

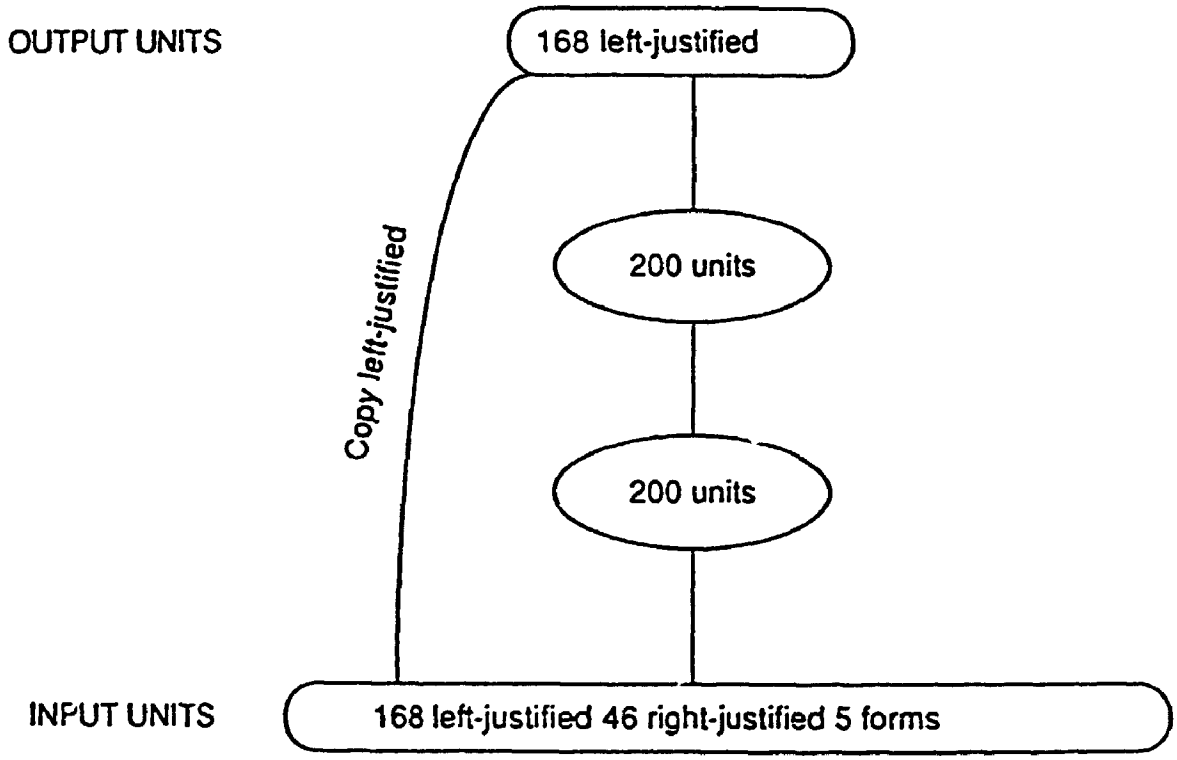
From MacWhinney and Leinbach's report one can infer that they used 1650 verbs for training -- 1532 regular and 118 irregular (cf. MacWhinney and Leinbach, 1991, p. 144). Training the network took 24,000 epochs. By epoch 16,000, all of the regular past tenses of the training sample were learned correctly. However, at the end of epoch 24,000 there were still 11 errors on the irregular pasts. This represents a 9.3% error rate in the irregular verbs used for training. MacWhinney and Leinbach believe that "if we had allowed the network to run for several additional days and given it additional hidden unit resources, we probably could have reached complete convergence" (MacWhinney and Leinbach, p. 151).

Training results *per se*, however, are meaningless if the network is not able to generalize from the previously "seen" examples to the "unseen" test examples. No matter how well trained, if the network is not capable of predicting correctly the past tenses of unseen verbs, there will be no justification for saying that the network has learned

Figure 3.3

MacWhinney and Leinbach's network.

(Reprinted from MacWhinney and Leinbach, 1991.


OUTPUT UNITS
168 left-justified
Copy left-justified
200 units
200 units
INPUT UNITS
168 left-justified 46 right-justified 5 forms

anything. Simply constructing an ANN-like look-up table that cannot predict the past tense of a new verb does not qualify as learning. Training a network that cannot predict anything new is similar to the operation of writing to memory in conventional computers -- we are as justified in calling the first operation "learning" as we are in giving that title to the second. Therefore, testing results on the unseen cases is crucial for gauging the success or failure of a learning model.

Surprisingly, despite the importance of test results for assessing the value of a learning model, MacWhinney and Leinbach tested the trained network on only 13 unseen irregular verbs. The result was that "9 of these untrained past tense irregulars were missed" (MacWhinney and Leinbach, 1991, p. 146). This represents 69.2 error rate on the irregulars. However, for no obvious reason they did not test their model on any of the unseen regular verbs: "Unfortunately, we did not test a similar set of 13 regulars" (MacWhinney and Leinbach, 1991, p. 151).

There are serious difficulties in estimating the significance of these results. Because the testing sample is so small -- only 11% of the irregular verbs in the training sample and a mere 0.8% of the whole training sample (!), and because we do not have any results on the unseen regulars (!), we are entirely in the dark regarding the overall error rate. Thus, despite the reported results we still do not know

anything truly significant about the inductive capabilities of MacWhinney and Leinbach's model.

Criticism of the MacWhinney and Leinbach Model

MacWhinney and Leinbach conclude that in light of the performance of their model "it is clear that the network succeeded in its assigned task of learning the English verb paradigm" (MacWhinney and Leinbach, p. 151) and thus, the success of their model supports a rule-less cue-based account of the verb inflection acquisition process (MacWhinney and Leinbach, p. 123). It is doubtful, however, that this conclusion is at all supported by MacWhinney and Leinbach's model or by the actual experimental results. The problems are mainly in three areas:

- The extent of the psychological reality of the model
- The extent to which TRICS have been tacitly used to bias the model in favour of a correct response
- The predictive success of the model

First, it has to be emphasised that MacWhinney and Leinbach's PDP model is very similar to the Rumelhart and McClelland model in that they use an ANN as a pattern associator in a supervised learning task. Both models share a common weakness in treating the learning of past tenses of a language as a totally isolated event. The ANN-based pattern associator has no access to lexical and/ or syntactic

information, it does not know what a verb is or what a past tense is. In particular, it is incapable of acquiring or representing morphological or lexical information in first-order form. MacWhinney and Leinbach appear to be unaware that they have provided the crucial categorial information during the supervised training process -- information which is otherwise unavailable in the natural learning environment of children -- and they misleadingly speak of their model as learning the past tenses of English verbs when they are at most justified in claiming that it has learned to associate phonetic patterns.

Second, a number of features are introduced without any apparent psychological justification. MacWhinney and Leinbach claim that they have improved the earlier Rumelhart and McClelland model by getting rid of the Wickelphone/ Wickelfeature representational format and thus have answered the many criticisms it entailed (cf. Lachter and Bever, 1988; Pinker and Prince, 1988). However, in MacWhinney and Leinbach's model, as with the Wickelphone format, phonetic information accessible to the network is not position independent. Instead of coding predecessor and successor phonemes as Wickelphones MacWhinney and Leinbach introduce special templates with which to code positional information. This means that the network is learning to associate patterns of phoneme/ positions within a predetermined consonant/ vowel pattern. The benefits of using a common consonant/ vowel grid

for all verbs are obvious. The network can "expect" to see only certain mini-patterns at certain places. And if fewer candidates compete for a certain position, the likelihood of a correct "guess" increases. It has an additional benefit that solves some intractable problems created by the Wickelphone/ Wickelfeature format. Thus, the confusions between similar words like slit - silt, or the impossibility of decoding correctly certain words -- e.g. the "algalgal" problem -- do not arise with the use of the consonant/ vowel template and the UNIBET format.

The important question, however, is whether there are viable psychological reasons for structuring all input and output with such a consonant/ vowel template. According to MacWhinney and Leinbach the psychological justification for using both left-justified and right-justified templates "derives from empirical work on language processing and acquisition that indicates that both children and adults pay attention to the beginning and to the end of the words" (MacWhinney and Leinbach, 1991, p. 142). Unfortunately, MacWhinney and Leinbach do not mention where the psychological justification for using the templates themselves derives from. The fact that it was easier for the network not to confuse similar sounding verbs using the template format certainly cannot count as a reason for adopting the format.

Another unjustifiable bias introduced with the choice of this format is in the types of verbs allowed in or rejected

from the training and the testing samples. Getting rid of all English verbs that have more than three syllables, more than three consonantal phonemes in a row, or more than two vocalic phonemes in a row, because they do not fit the chosen template, biases the model in favour of shorter verbs, predominantly of Anglo-Saxon origin, against longer verbs, predominantly composite, of Latin and French origin.

The absence of a psychological justification is not confined only to the selection and representation of data. Several questions hang over the architecture of the network itself. First, it is not clear why each unit in the input and output layers was chosen to represent a single phonetic feature like front, centre, back, high, etc. What is it that makes individual neurons capable of representing only phonetic features, but not phonemes, whole words, or finer grained "microfeatures"? There is no discussion of the reasons for this particular choice of coding; MacWhinney and Leinbach could have used arbitrary binary coding with at least $\log_2 N$ bits, where N is the number of phonemes. Such varying of the coding could have shown whether the choice of coding can introduce favourable biases.

But there is another particularly puzzling architectural detail in MacWhinney and Leinbach's model: the set of identity-mapping connections between the left-justified input and the output. We saw that this feature obviously facilitates the learning of the verbs that change from present to past in

a regular fashion, i.e. the majority of English verbs. According to MacWhinney and Leinbach this is an essential feature of the network, and it is psychologically justifiable:

> The network was designed to treat the learning of the "derived" forms of the verb as modifications of the phonological form of the "basic" present tense. The idea is that the child assumes that the past tense is somehow a modification of the present. This is done by including a "copy" of the left-justified phonological form of the input directly onto the output. (MacWhinney and Leinbach, 1991, p. 143.)

It is quite clear that this copying device is a TRIC. The only thing that is not clear is why the child needs to have this modification assumption hardwired in his brain, and at the same time lacks other much more reasonable assumptions, such as the assumption that most verbs' past tenses take the verb stem and add /Id/, /d/, or /t/ to the end. After all, when it comes to TRICS, if the second assumption is hardwired in the network, it will probably achieve better results. We suppose that the first assumption was chosen over the second which is by far a more reasonable assumption, because it does not look like a rule. The appearance, however, can be misleading:

> If x is a verb stem, then x is going to be slightly modified in its past tense form

is a rule, even though it is one that does not capture a significant generalization.

Generally, it is very difficult to assess the actual performance of PDP models because there are so many parameters (different initial weights, number of hidden units, number of hidden layers) that have to be tuned by hand during many unsuccessful attempts in order to adapt the network for a specific application. It is unclear, how this methodology can affect the inductive capacities of the neural networks, since so many choices -- like MacWhinney and Leinbach's copying mechanism, for example -- appear to be made specifically for the purpose of enabling the ANN to overcome a recalcitrant obstacle. These ad hoc choices may (unbeknownst to the investigators) amount to a complex procedure to create a network that "fits the data". So far, the problem of distinguishing networks designed to be application-specific from general-purpose neural networks has not received sufficient attention in the literature.

But my third and main criticism is directed at the actual performance of MacWhinney and Leinbach model. A learning model which has a very poor psychological justification might still be interesting -- if only from the point of view of applications -- if, despite its conceptual flaws, it has managed to achieve good performance results. Unfortunately, the performance of MacWhinney and Leinbach's model is at best very unclear and at worst outright disappointing. A model without a clearly assigned predictive accuracy is as good as a model with 0% predictive accuracy. Guessing correctly 4 out

of 13 test verbs cannot even begin to reveal the realistic error (success) rate of the model because of the extreme unreliability of the testing sample which was used to measure it. As we saw MacWhinney and Leinbach's testing sample consists of only 11% of the irregular verbs in the training sample and only 0.8% of the total training sample. Worse still, MacWhinney and Leinbach did not test any regular verbs (cf. MacWhinney and Leinbach, 1991, p. 144, and p. 151). But one of the main things that we need to know about their learning model is whether it can learn the past tenses of the majority of English verbs -- the regular past forms. Moreover, the 13 test verbs were not randomly chosen. No attempt was made to vary the ratio between training and testing samples in a series of different learning experiments. This indicates that MacWhinney and Leinbach did not seriously attempt to test the inductive capabilities of their model. Without such a test, the empirical evidence they report cannot support their theoretical claims, in fact it cannot support any theoretical claims.

MacWhinney and Leinbach appear to be unconcerned with the extremely unreliable testing sample and the low generalization capacity of the model reported on the basis of this sample. Their discussion of the main achievements of their model on pp. 146-53 is based mainly on results obtained from the training process! They seem to believe that simply because the network was trained to associate certain phonetic patterns,

the network has learned to produce the past tenses of English verbs. For example they write:

> Performance on the regular past tense was perfected about midway through the simulation. By the first test point [i.e. the first check during the training process on the training verbs] it was already over 99% correct. In overall terms, the simulation did very well at its task of learning the past tense. (MacWhinney and Leinbach, 1991, p. 146.)

MacWhinney and Leinbach are convinced their simulation did well without actually checking whether it could correctly produce correctly even a single past of an unseen regular verb!

This betrays a deep misunderstanding of one of the most basic principles in traditional machine-learning studies; what is important in a machine learning experiment, especially one that is claimed to be relevant for cognitive science, is not how well the learning program (or the network) is trained, the crucial measure is how well this program (or network) performs after it has been trained, i.e. how well it is able to predict the right answers to problems it has never previously encountered. Alternatively, MacWhinney and Leinbach could have used the PAC (Probably Approximately Correct) learning paradigm (Valiant, 1984) in which training and testing samples are drawn randomly according to some fixed distribution over the whole sample space. The passing of any

such elementary testing procedure makes all the difference between a learning system that is capable of inductive generalization and a system that acts as a complicated look-up table (connectionist or otherwise), and thus "knows" all the answers to the questions it was designed to answer, but lacks any generative capacity.

The analysis of MacWhinney and Leinbach's model leads to the conclusion that they have failed to provide a successful model for learning the past tenses of English verbs. There is no indication that they have produced any results for past participle, present participle, and 3d person singular, so their claim that the model has succeeded in learning the English verb paradigm is absolutely unfounded. In many respects MacWhinney and Leinbach's model does not represent an improvement over Rumelhart and McClelland's model and in some respects, e.g. in overall error rate and in the openness of the reporting of the errors of their model, it actually represents a backward step. It must be rejected as a model of the competence of native speakers' acquisition the past tenses of English verbs for the same reasons that led to the rejection of Rumelhart and McClelland's model:

- the isolation of the past tense acquisition process from the overall process of language acquisition,
- dissociating of phonetic from lexical and morphological information,
- mistaking the association of chosen phonetic patterns

with the induction of the category "past tense" and the rules for its application,

- use of psychologically unwarranted TRICS,
- failure to represent the acquired phonetic knowledge in usable/ interpretable form,
- extremely poor performance results, i.e. no testing on regulars, minute test sample, no random sampling, no estimate of overall error rates, no analysis of typical mistakes, etc. (In some respects this represents a step back from Rumelhart and McClelland's model.)

Given its conceptual shortcomings and its uncertain and poor performance, this model can hardly serve as an example of how to eliminate rules and symbolic representations in the explanation of our knowledge of language.

Is There a Better Symbolic Model?

I entirely agree with MacWhinney and Leinbach when they say that conceptualizations (theories) should ultimately be accepted or rejected in view of the success or failure of their implementations (experimental predictions). I also agree with them when they say that

> If there were some other approach that provided an even more accurate characterization of the learning process, we might still be forced to reject the connectionist approach, despite its successes. The proper way of debating conceptualizations is by contrasting competitive

> implementations. To do this in the present case, we would need a symbolic implementation that could be contrasted with the current implementation. (MacWhinney and Leinbach, 1991, p.153)

But for the theoretical reasons outlined earlier and because of the absence of convincing experimental evidence, I do not agree that either Rumelhart and McClelland or MacWhinney and Leinbach have provided an adequate learning model for the acquisition of the past tenses of English verbs. They were right in saying that so far there were no rival symbolic implementations that could achieve even the limited performance results of the connectionist models. But in a recent report Ling and Marinov (1992) take up MacWhinney and Leinbach's challenge and provide a symbolic learning model that can learn the past tenses of English verbs much better than any PDP model so far. I will demonstrate that the success of Ling and Marinov's (1992) symbolic learning model should lead eventually to the reevaluation of most of the evidence in favour of the eliminativist approach to language learning and language processing in general.

The Symbolic Pattern Associator

The core of Ling and Marinov's model is the Symbolic Pattern Associator (SPA for short). The SPA is a very general and very efficient symbolic algorithm for associating

arbitrary patterns which have a finite number of components with a finite number of discrete values. So far, there is no evidence to suggest that ANNs have any systematic advantage over the SPA as pattern associating and pattern recognition devices. On the contrary, in many respects exactly the opposite is true.

The Requirements for the Model

It is not difficult to notice that most of the publicity that connectionist research has achieved is due to the fact that connectionist systems are capable of supervised learning from examples. The main connectionist learning algorithm -- the error backpropagation algorithm -- as well as the similar perceptron convergence algorithm, enable connectionist networks to learn to associate arbitrary boolean and/ or numerical vectors (patterns) and in some cases to generalize successfully to new unseen patterns, i.e. given an input pattern not encountered during the supervised training to produce the "correct" or desired output pattern without any supervision.

One common feature that all connectionist networks share is that they have great difficulty accounting for knowledge representation. In connectionist research only the activations of input and output units receive direct interpretation, the so called hidden units and the connection strength matrices that map the input patterns of activation to the output

patterns cannot be interpreted directly; they are in fact completely opaque from the point of view of knowledge representation. This problem has given rise to two general schools of connectionist thought -- eliminative connectionism and implementational connectionism. As we saw earlier (in Chapter Three) eliminative connectionists, like Rumelhart and McClelland, maintain that human cognition can be explained without any appeal to explicit inaccessible rules or to symbolic representations. They believe that the language of patterns of activation and connection strengths will suffice. Accordingly, there is no problem about knowledge representation: cognition can be explained simply as a pattern association and pattern recognition process; there is therefore no need to have a direct interpretation of the associating medium.

There is a problem with this connectionist position. On the one hand, because of its failure to account for knowledge representation, eliminative connectionism is a doctrine that represents a throwback to the old days of associationism and behaviourism. On the other hand, it is easy to see the attractiveness of having a learning system that can learn to associate arbitrary patterns, and extract whatever regularities are present in the data. At the same time, a learning system that is incapable of representing the acquired knowledge has virtually no value compared with a learning system that can represent the knowledge it has acquired. That

is why a symbolic system capable of challenging any eliminative connectionist model has to meet two conditions:

1. It must be able to match fully the learning capabilities of neural networks, i.e. it must be capable of supervised learning from examples.
2. Unlike neural networks, it must be able to represent the knowledge acquired in the learning process.

Many connectionists believe that neural networks possess some unique advantage over symbolic systems in that they are capable of learning things that symbolic systems are inherently incapable of learning. As we saw earlier (in Chapter Two), it is also quite common to read that symbolic systems are 'hard' and 'brittle', while connectionists systems are 'soft' and 'flexible', and can account for a wider range of cognitive phenomena. Such beliefs are widespread because many people are not aware of the fact that there are several symbolic learning algorithms that can compete quite successfully with ANNs on a wide range of practical learning tasks, even with such successful connectionist simulations like NETtalk (Atlas et al., 1990; Shavlik, Mooney, and Towell, 1991; Marinov, 1992). One of the most widely studied of these systems that can match the learning abilities of ANNs and at the same time can represent explicitly the acquired knowledge is ID3 (Quinlan, 1986a) which we described earlier (in Chapter Two).

The Architecture of the Symbolic Pattern Associator

Although, ID3 and C4, as well as other TDIDT symbolic algorithms, have been very successful in challenging the learning and the representational capabilities of neural networks (cf. Chapter Two) they are essentially limited in the type of learning tasks that they can perform. If the learning task is to learn to classify a set of different patterns into several mutually exclusive categories (as many of the most successful connectionist simulations actually do) ID3 and C4 have been shown to perform as well as, or even better than neural networks. However, if the task is to classify a set of patterns into many possibly partially overlapping patterns, the neural networks retain a distinct advantage. The reason is that these are typical pattern association tasks, tasks to which ID3 and C4 as exclusive classifiers are not particularly well-suited. The best ID3 and C4 can do is to treat the different output patterns as mutually exclusive classes which usually result in exponential growth and loss of generalization capacity.

In order to rival fully the learning capabilities of ANNs the powers of ID3 and C4 and similar TDIDT systems have to be increased and they have to be turned into general purpose symbolic pattern associators. Ling and Marinov do this by combining the power of individual decision trees into a "forest" or set of trees all of which work together to accomplish the task of associating two (sets) of arbitrary,

possibly overlapping, patterns. In their Symbolic Pattern Associator each tree takes as input the same set of input patterns that is available to all the other trees; but each tree is concerned with determining only a portion of the output pattern, usually the value of only one of its attributes. The set of trees associating the two sets of patterns can be built serially or in parallel since each tree is built independently from the others. A parallel implementation can lead to great gains in speed.

In order to understand how this General Purpose SPA works, let us look at a simple example. Suppose, we want to associate in a one-to-one fashion two sets of arbitrary patterns that have n attributes each, where each attribute has binary values 1,0 (any non-binary discrete values are also possible). (See Table 3.2.) The general purpose SPA is able to map all the patterns in the first set (i.e. the input patterns) one-to-one onto the patterns in the second set (i.e., the output patterns). After training, given the input pattern IP_1, the SPA will produce the "correct" output pattern OP_1; given the input pattern IP_2 it will produce the "correct" output pattern O^{Γ} , etc. The way the SPA achieves this is by building a tree that takes all input patterns and classifies them with respect to the values of the first output attribute Ω_1, then a second tree takes again all input patterns and classifies them with respect to the values of the second

Table 3.2

Binary Patterns to Be Associated by the SPA

Input patterns						Output patterns					
Pat.	Attributes					Pat.	Attributes				
no.	I_1	I_2	I_3	...	I_n	no.	Ω_1	Ω_2	Ω_3	...	Ω_n
IP_1	1	0	1	...	0	OP_1	1	1	0	...	1
IP_2	0	1	0	...	1	OP_2	1	1	1	...	0
IP_3	0	1	0	...	0	OP_3	1	0	1	...	1
	.	.	.				.	.	.		
IP_n	.	.	.		.	OP_n	.	.	.		.

Table 3.2

Binary Patterns to Be Associated by the SPA

Input patterns						Output patterns					
Pat.	Attributes					Pat.	Attributes				
no.	I_1	I_2	I_3	. . .	I_n	no.	Ω_1	Ω_2	Ω_3	. . .	Ω_n
IP_1	1	0	1	. . .	0	OP_1	1	1	0	. . .	1
IP_2	0	1	0	. . .	1	OP_2	1	1	1	. . .	0
IP_3	0	1	0	. . .	0	OP_3	1	0	1	. . .	1
	.	.	.				.	.	.		
IP_n	.	.	.		.	OP_n	.	.	.		.

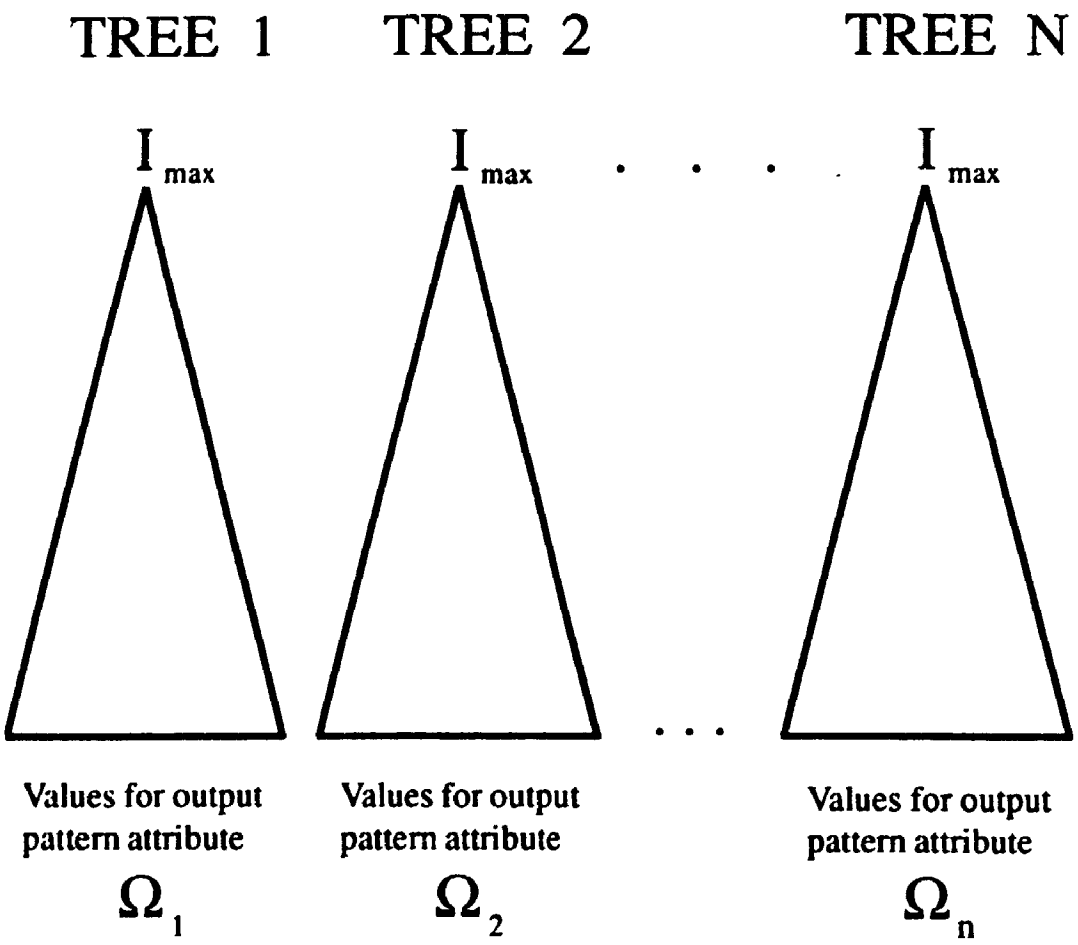
output attribute Ω_2, and so on until all output attributes are exhausted. The resulting trees constitute a joint decision method (see Figure 3.4) for associating each (training) input pattern to the correct output pattern with 100% accuracy unless there is contradictory information in the data, i.e. unless one and the same input pattern is mapped twice on two entirely different output patterns. Given what we know about the TDIDT algorithms, this is not surprising at all.

What is very interesting, however, is that once the SPA has been trained, i.e. once the set of associating trees has been built, it can have remarkable predictive accuracy on "unseen" pairs of patterns. It can extend the knowledge acquired during the learning process, and it can generalize from the training examples to unknown examples. Obviously, if the SPA can accomplish such a high degree of predictive accuracy, it must be able to extract some of the regularities in the training sample, and then project this knowledge onto future cases. In effect, the SPA can rival the learning capabilities of any artificial neural network using a supervised learning algorithm, and capable of learning to generalize from a set of examples to new "unseen" cases.

In addition to these remarkable learning powers, the SPA has an easy way of representing the knowledge that it has acquired. The decision tree format is already a useful form of knowledge representation. Moreover, the SPA can automatically convert its decision trees into sets of

Figure 3.4

SPA: A joint decision method for associating arbitrary patterns.


TREE 1
TREE 2
TREE N
I max
I max
I max
Values for output pattern attribute
Ω 1
Values for output pattern attribute
Ω 2
Values for output pattern attribute
Ω n

production rules. These production rules can be meaningful to the human observer, if there is a meaningful connection between the associated patterns.

To summarize: The main advantages of the SPA vis-a-vis ANNs using supervised learning algorithms are threefold:

- In pattern association tasks, where each pattern consists of a finite list of features whose values are finite and discrete, there is absolutely no advantage at all to using ANNs over SPA in terms of accuracy of learning of the training examples, inductive generalization abilities, (correctly predicting new "unseen" cases) or speed of learning (for serial and/ or parallel implementation).
- The SPA does not use any TRICS and requires no parameter tuning.
- In different learning systems in which everything else such as accuracy of learning, predictive success, speed of learning, etc. are equal, the learning systems that are able to produce explicit representations are certainly better than the ones which lack this capacity. Explicit representations allow for the possibility of further processing of the acquired knowledge and for the flexible integration of this knowledge in various domains. It is unclear how learning systems that "represent" knowledge in "black boxes" can coordinate, combine, or further generalize their knowledge.

Experimental Set-Up

Ling and Marinov structured their experiment in such a way as to guarantee a maximal common basis for comparison. Since MacWhinney and Leinbach used a much larger set of verbs than Rumelhart and McClelland, Ling and Marinov borrowed the list of verbs that MacWhinney and Leinbach used in their learning experiment. It contained exactly 1404 present tense/ past tense verb form pairs. (Actually, there appears to be a minor discrepancy between the two lists. According to their report, MacWhinney and Leinbach used a slightly larger set of training verbs -- 1650.) Because of the large overlap in the verb sets used in the two experiments, Ling and Marinov's results can be directly compared with MacWhinney and Leinbach's results. Even though Rumelhart and McClelland used altogether only 506 verbs, the overlap between the sets is still sufficient for a similar comparison.

There are, however, several significant changes from MacWhinney and Leinbach's, and Rumelhart and McClelland's, experimental set up:

First, Ling and Marinov eliminated the psychologically unjustified template/ feature representational format and used simple left-to-right phonetic coding in the UNIBET format. (They also carried out an additional test using the template format to see if that would change significantly their results -- it did not. They recorded only a 2.6% decrease in overall accuracy due to the use of the templates.) They did not

attempt to use Wickelphone/ Wickelfeature format since it was shown to be psychologically and linguistically unjustified by Pinker and Prince's and Lachter and Bever's criticisms. Also, unlike MacWhinney and Leinbach's model they did not include any special purpose copying devices that would allow for a verb stem to influence directly the output of their system.

Second, in order to guarantee unbiased samples and a clear and unambiguous reading of the results, Ling and Marinov decided to use only randomly drawn mutually disjoint training and testing samples in several independent training and testing trials. They used a program that randomly selects a specified number of verbs and removes them from the original list. This allowed them, in contrast to MacWhinney and Leinbach and Rumelhart and McClelland, to vary the relative sizes of the training and testing samples at different trials, and thus achieve a robust estimate of the generalization abilities of the SPA.

Third, Ling and Marinov present their main result in the form of a basic estimate of the capacity of their model to learn the past tenses of English as closely as possible to the level of adult competence. They did not attempt to model the U-shaped learning curve of the acquisition of the English past tense. In order to achieve this basic estimate they did not attempt to present the model with frequency information for the different verbs. Rumelhart and McClelland's way of using frequency information in order to achieve the U-shaped curve

was discredited by Pinker and Prince's, Lachter and Bever's, and Plunkett and Marchman's criticisms. MacWhinney and Leinbach traced the learning rate for only some limited periods of epochs and report that although some U-shaped learning was observed, most of the training time of the network was spent in error free performance. There is no question that an early exposure to certain irregulars, followed by increased exposure to regular verbs, would have created a U-shaped curve in Ling and Marinov's experiment. However, the psychologically justifiable rate of supply of regulars and irregulars during training is still unclear. But the main reason for presenting every verb to the SPA only once is that this gives a robust estimate of the system, irrespective of the frequency variations in different linguistic environments. Linguistic environments for all natural languages vary, and verbs that are now used frequently in one English-speaking community may not have been used as frequently in the past, or in other communities. Yet children always succeed in learning the past tenses of even infrequently occurring verbs. The goal is to provide a model of the basic capacity to acquire inflectional forms.

As it happened the verbs in MacWhinney and Leinbach's list were not longer than 10 phonemes so they provided 15 input and output attributes, i.e. the SPA in this particular experimental set-up could not handle words longer than 15 phonemes, although, if any such words were found the size of

the input and output patterns could easily be increased. The values for each attribute ranged over all phonemes in the UNIBET format. In contrast to MacWhinney and Leinbach who separated the processing of consonants from the processing of vowels without any justification, Ling and Marinov make no additional assumptions that could be classified as TRICS, and used all phonemes as values for all attributes, even though the other approach would have simplified significantly the trees built during learning, and would have increased the speed of the system.

Each trial consisted of one training and one testing session. During the training process, the pattern associator had to look at the set of training examples of correct stem/ past tense verb pairs (see Table 3.3), and then build a set of trees that associates them in one-to-one fashion with 100% accuracy. Since there were only 15 attributes in each training session the SPA produced only 15 trees that were able to associate all known pairs of verb forms correctly and produce answers to the "unseen patterns".

Ling and Marinov observed that the SPA discovered very easily MacWhinney and Leinbach's "assumption" that usually English verbs change very little from present to past tense, i.e. most of the branches of its trees were inactive, they just passed the same value of the examined attribute to the terminal leaf. All trees had a similar overall structure with

Table 3.3

Pairs of Phonetic Patterns for Training

Present tense form										Past tense form									
Phoneme attributes										Phoneme attributes									
I_1	I_2	I_3	I_4	I_5	.....	I_{15}				Ω_1	Ω_2	Ω_3	Ω_4	Ω_5	.....	Ω_{15}			
b	E	t	$	$	.....	$				b	E	t	$	$	.....	$			
p	6	r	s	i	v	$	$	..	$	p	6	r	s	i	v	d	$	$	. $
r	3	d	$	$	.....	$				r	o	d	$	$	.....	$			
I	n	t	E	n	d	$	$	..	$	I	n	t	E	n	d	I	d	$	. $
p	r	E	s	$	$	.....	$			p	r	E	s	t	$	$	.....	$	
		.	.	.									.	.	.				

Table 3.3

Pairs of Phonetic Patterns for Training

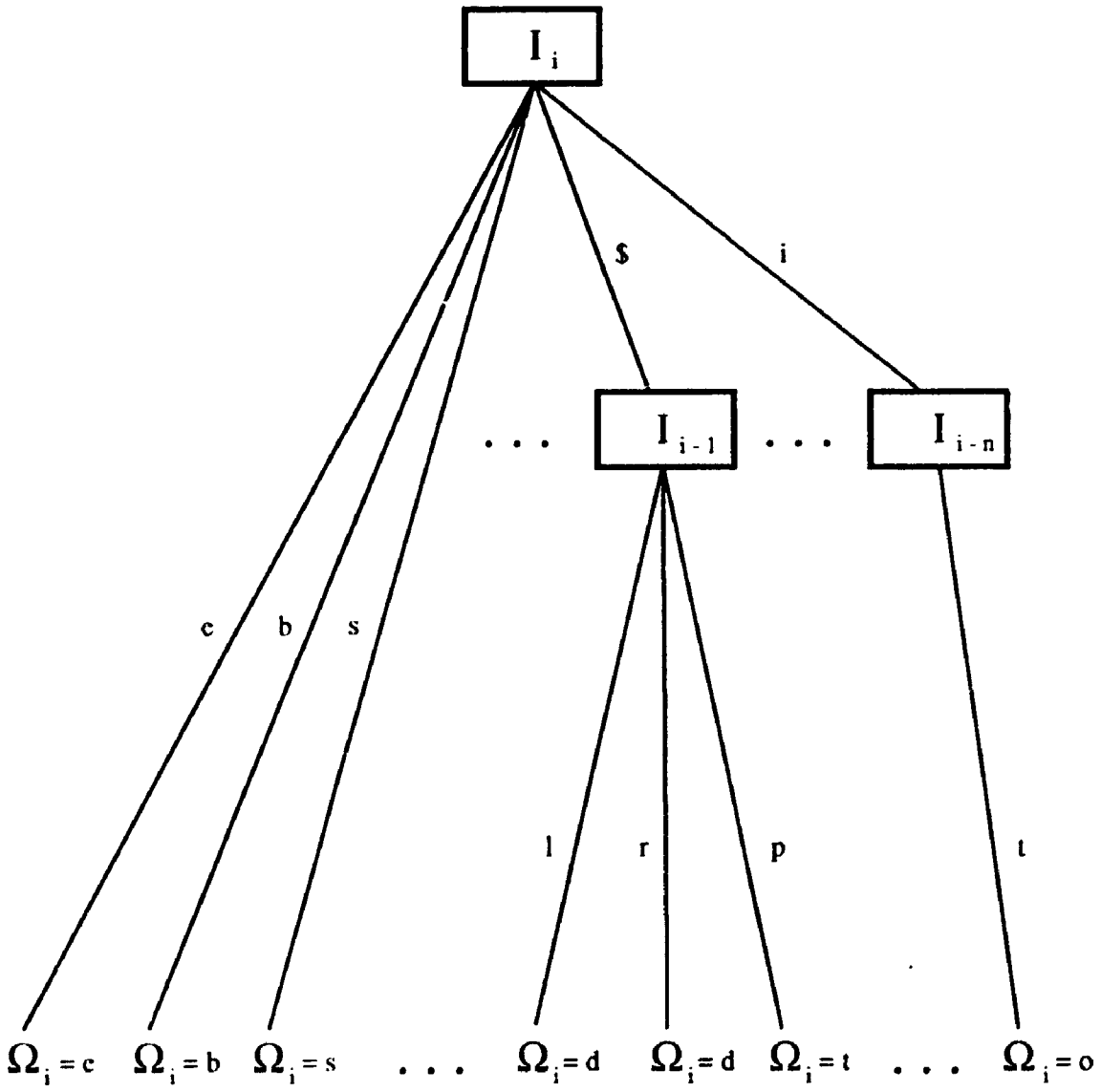
Present tense form															Past tense form														
Phoneme attributes															Phoneme attributes														
I_1	I_2	I_3	I_4	I_5	.....	I_{15}									Ω_1	Ω_2	Ω_3	Ω_4	Ω_5	.....	Ω_{15}								
b	E	t	$	$	.....	$									b	E	t	$	$	.....	$								
p	6	r	s	i	v	$	$	....	$						p	6	r	s	i	v	d	$	...	$					
r	3	d	$	$	$	$	$	....	$						r	o	d	$	$	$	$	$	...	$					
I	n	t	E	n	d	$	$	....	$						I	n	t	E	n	d	I	d	$	.	$				
p	r	E	s	$	$	$	$	....	$						p	r	E	s	t	$	$	$	$	.	$				
.	.	.													.	.	.												

many identical branches and subbranches. (See Figure 3.5.) All branches (on all trees) can broadly be grouped into two main clusters: identity preserving branches and changing branches. The changing branches, on the whole formed two distinguishable subgroups: one group was concerned mainly with examining these phoneme positions where a phoneme was followed with a blank $ sign, these branches resulted usually in the assignment of /d/ or /t/ or /Id/ to the output pattern. If certain vowels were encountered, preceding positions were checked, and if "exceptional sounding" phonemes were discovered, the tree branched further to accommodate the exceptional irregular forms. Thus, most of the trees have a similar structure with many trees "sharing" isomorphic branches.

Obviously, the SPA was able to discover which positions and which phonemes in the verb stems were most relevant for the production of the correct past tense forms. For example, it discovered that the phonemes at the end of the verbs (phonemes followed by a blank $ sign) usually control the addition of /d/, /t/, or /Id/ in the past tense form. It also tried to accommodate the numerous exceptions in the formation of irregular past tenses. One has to be careful, however, to note that the information available to the tree is always <u>position dependent phonetic information</u>. The trees do not have direct access to categories like "verb stem", "suffix", "past tense", etc.

Figure 3.5

Typical phonetic decision tree created during the learning process.


I_i
$
i
I_{i-1}
I_{i-n}
c
b
s
l
r
p
t
$\Omega_i = c$
$\Omega_i = b$
$\Omega_i = s$
$\Omega_i = d$
$\Omega_i = d$
$\Omega_i = t$
$\Omega_i = o$
Identity preserving branches
Suffix adding branches
Exception branches

Performance Of the Symbolic Pattern Associator On the Past Tense Learning Task

Ling and Marinov obtained from MacWhinney and Leinbach's original list 1404 correct present tense/ past tense pairs of verb forms; these pairs contained 1263 regular, and 141 irregular, past tense forms. They carried out a total of 12 trials with different randomly selected disjoint training and testing samples of 4 different sizes. Learning of the training examples was 100% successful. The average rate of correctly produced past tenses that were not encountered during training was 78.76% of the testing samples with a top result of 82.7%! (In all cases past tenses which differed even with a single phoneme from the correct output were counted as errors. For the record, Rumelhart and McClelland's overall correct prediction rate was only 57%. MacWhinney and Leinbach did not provide any results that could test the overall generalization capacity of their model. Note also that the average user time on a MIPS machine for all 12 trials was only 1:22 min. compared with the extremely slow connectionist models that took days, even weeks to complete a single learning trial. Table 3.4 summarizes Ling and Marinov's results. (See Table 3.4.)

An important point of contention in the criticism of the connectionist models concerns the predominant types of errors the connectionist networks were actually making. Rumelhart and

Table 3.4

Results of the Performance of the SPA on the Learning of the Past Tenses of English Verbs

Trial no.	Test/ training sample (verbs)	Time (min)	Accuracy during training (%)	Accuracy during testing (%)
1	500/500	0:39	100	76.0
2	500/500	0:39	100	77.2
3	500/500	0:42	100	77.6
4	500/700	0:59	100	78.4
5	500/700	0:58	100	78.0
6	500/700	0:52	100	80.6
7	300/1000	1:40	100	82.7
8	300/1000	1:40	100	79.3
9	300/1000	1:41	100	76.3
10	100/1280	2:12	100	78.0
11	100/1280	2:10	100	80.0
12	100/1280	2:16	100	81.0
Average	350/870	1:22	100	78.76

McClelland were remarkably open in discussing the incorrect responses of their model. Unfortunately, MacWhinney and Leinbach do not provide any information about the types of errors their model committed, apart from saying that four of the 13 irregular test verbs were regularized. Thus, we lack any information about the other errors their model made on the irregulars and no information whatsoever on the regulars. That is why Ling and Marinov decided to follow the open approach of Rumelhart and McClelland and to give full information about the errors made by the SPA.

Ling and Marinov divided all possible types of incorrect response into five categories: regularization errors (treating an irregular past as regular); no change errors (the regular or changing irregular verb was treated as a no change irregular); vowel change errors; consonant change errors (these apply if there was only one error in the word); and "impossible errors" (any other errors).

In order to minimize possible biases in the small samples used in the connectionist models, Ling and Marinov measured the distribution of errors on a randomly chosen test sample of 600 verbs. The SPA was trained on another randomly chosen disjoint sample of 700 verbs. (They again followed the traditional machine learning approach and they did not use training verbs in the testing sample.) The results are summarised in Table 3.5. (See Table 3.5.)

Table 3.5

Types of Responses

Verb type	Percent of sample	Correct (%)	Incorrect (%)				
			Reg.	No-chg.	V-chg.	C-chg.	Imp.
Irregular	7.8	27.7	53.2	14.9	2.1	0	2.1
Regular	92.2	86.8	n/a	4.8	3.2	2.3	2.7
Total	100.0	82.2	4.2	5.6	3.2	2.2	2.6

As can be seen from Table 3.5, the incorrect responses were disproportionally concentrated in the irregular verb subset. This is exactly what the traditional rule-based account leads us to expect, since the system has no access to morphological or lexical information at this stage and relies only on phonetic information; indeed, it is surprising that it managed to predict correctly even 27.7% of the irregulars. For the record, the success rate of the SPA on the irregulars still compares very well with the 7.1% success rate for Rumelhart and McClelland's model but is slightly lower than the 30.8% success rate of MacWhinney and Leinbach's model. Ling and Marinov's result, however, is much more precise, since it was achieved when they tested a total of 47 randomly chosen irregulars; by contrast, MacWhinney and Leinbach tested only 13 specifically chosen irregulars.

One reason for the large number of errors on irregular pasts is that the SPA had no way of distinguishing between homonyms. In many cases, perfectly good productions like ring --> ringed (in the sense of encircle) were treated as errors, e.g. as the mistake on the production ring --> rang (in the sense of resound). Also, according to the rule-based account of language acquisition, exception markedness plays a very important role in learning the irregular pasts. Such information, however, is not available at the phonetic level.

Of course, Ling and Marinov could have easily coded such non-phonetic information into their data -- one can always add

one or two extra attributes that can denote morphological category, exceptional subgroup, etc. This is the approach taken by MacWhinney and Leinbach when they want to teach their network to learn the full verb paradigm of English. Thus, MacWhinney and Leinbach coded in the input template the information about whether the network has to produce past tense, past participle, present participle, or third person singular. This is an obvious TRIC that lacks any psychological justification -- the network is receiving additional non-phonetic categorial information in pseudo-phonetic form. The human auditory perceptual system certainly does not have access to such information.[5] In light of the fact that the SPA was exposed only to different phoneme patterns during training, it is remarkable that it was able to predict 27.7% of the unseen irregulars correctly. This shows that even in irregular pasts there are certain limited recurring regularities that can be extracted by the SPA.

Much more interesting for this investigation, however, is the distribution of the different types of incorrect responses. It is evident that regularization responses are by far the major source of error in the case of irregular verbs. In the absence of additional non-phonetic exceptional information, the SPA overgeneralizes in favour of the predominant regular past formation. Compared with the correct and regularization responses the other types of errors, especially the "impossible" errors, appear to be

insignificant.

Ling and Marinov follow Rumelhart and McClelland (but not MacWhinney and Leinbach) in listing some of the errors on the irregular and regular verbs drawn at random from the test sample. They also included some examples of correctly produced irregulars to demonstrate the difficulty of some of the forms (i.e. it appears to be relatively easy to produce a no change correct irregular but it is not so easy to get a correct vowel change irregular):

Regularization errors: (The incorrect forms produced by the SPA are marked with *; the correct forms as classified by MacWhinney and Leinbach are shown in parentheses.)

bid /bid/ -- *bidded /bidId/; (bid /bid/)

shine /S3n/ -- *shined /S3nd/; (shone /Son/)

flee /fli/ -- *fleed /flid/; (fled /flEd/)

ring /rIN/ -- *ringed /rINd/; (rang /r&N/)

seek /sik/ -- *seeked /sikt/; (sought /sOt/)

...

No change errors:

blast /bl&st/ -- *blast /bl&st/; (blasted /bl&stId/)

raise /rez/ -- *raise /rez/; (raised /rezd/)

absorb /6bsOrb/ -- *absorb /6bsOrb/; (absorbed /6bsOrbd/)

fold /fold/ -- *fold /fold/; (folded /foldId/)

spring /spriN/ -- *spring /spriN/; (sprang /spr&N/)

...

Vowel change errors:

say /se/ -- *sayd /sed/; (said /sEd/)

speed /spid/ -- *speded /spEdId/; (speeded /spidId/)

chew /tSu/ -- *chowed /tSod/; (chewed /tSud/)

rumble /r6mb6l/ -- *rambled /r&mb6ld/; rumbled /r6mb6ld/

jingle /dZINg6l/ -- *junq..ed /dZ6Ng6ld/; (jingled /dZINg6ld/)

...

consonant changes:

shy /S3/ -- *shied /S3t/; (shied /S3d/)

lie /l3/ -- *lied /l3t/; (lied /l3d/)

constitute /kanst6tut/ -- *constided /kanst6tudId/;
(constituted /kanst6tutId/)

graduate /gr&dZ6w6t/ -- *graduared /gr&dZ6w6rId/;
(graduated /gr&dZ6w6tId/)

interpret /Int6rpr6t/ -- *interprerid /Int6rpr6rId/;
(interpreted /Int6rpr6tId/

...

impossible errors:

prescribe /prIskr3b/ -- *prescry /prIskr3/;
(prescribed /prIskr3bd/)

characterize /k&rIkt6r3z/ -- *charactered /k&rIkt6rd/;
(characterized /k&rIkt6r3zd/)

recommend /rEk6mEnd/ -- *recomed /rEk6md/;
(recommended /rEk6mEndId/)

preserve /prIz6rv/ -- *preser /prIz6r/;

(preserved /prIz6rvd/)

transform /tr&nsfOrm/ -- *transford /tr&nsfOrd/;

(transformed /tr&nsfOrmd/)

...

correct productions:

find /f3nd/ -- found /f1nd/

ride /r3d/ -- rode /rod/

stride /str3d/ -- strode /strod/

sting /stIN/ -- stung /st6N/

fling /flIN/ -- flung /fl6N/

sweep /swip/ -- swept /swEpt/

sleep /slip/ -- slept /slEpt/

spread /sprEd/ -- spread /sprEd/

set /sEt/ -- set /sEt/

bet /bEt/ -- bet /bEt/

...

Most of the "impossible" errors were a result of the majority default heuristic used in C4 and incorporated in the SPA as well: During training if there is no instance falling under one branch, assign the default class to the literal of that branch, where the default class is the one with the maximum number of positive instances. This created a problem with longer verbs where blanks which become the default class in the last several trees were sometimes inserted, thus truncating the past tense form of longer verbs. This default heuristic can be changed to better suit the particularities

of the verb past tense formation and thus significantly reduce the number of "impossible" mistakes. Such a change, however, would make the SPA application-specific, and this is not desirable if we want to observe how a general purpose pattern associator can perform on this learning task.

It is surprising how good the performance of the SPA is. Given the fact that during testing, the SPA can use only local phonetic information (it has no access to lexical or morphological information), it performed exceptionally well, far surpassing the performance of the connectionist models. A brief look at Table 3.6 which compares results achieved by the SPA and the results of the other two connectionist models shows the superior performance of Ling and Marinov's model. (See Table 3.6.)

The results presented in Table 3.6 are a convincing proof that the SPA pattern associator outperforms MacWhinney and Leinbach's, as well as Rumelhart and McClelland's connectionist models in the task of learning the past tenses of the English verbs. It would be interesting to see if they or any other researchers could construct a PDP model that can match the learning and inductive generalization abilities of the SPA in the task of learning the past tenses of English verbs or any other learning task.

However, I do not consider the performance of the SPA (and by implication of connectionist systems) to be alone sufficient to explain how children acquire the English past

Table 3.6

A comparison between SPA and the Connectionist Models in Learning of the Past Tenses of English Verbs

	R&M	M&L	SPA
Test sample as percentage of training sample	20	0.8	50-100
Random samples	No	No	Yes
Results averaged over multiple trials	No	No	Yes
Accuracy on training sample	97-98	99.3	100
regulars	98	100.0	100
irregulars	95	90.7	100
Accuracy on testing sample	57.0	N/A	78.8
regulars	66.7	N/A	86.8
irregulars	7.1	30.8	27.7
Average learning time	280 hours	days/ weeks(?)	6.7 min

tense. What the SPA does establish is how information about purely phonetic regularities can be extracted and expressed in the form of production rules; this opens the possibility of these rules generalizing and integrating with lexical, morphological, and syntactic rules to form a model of the competence of the mature speaker.

Explicit Representation and Higher-Level Processing

The SPA successfully met the first of the two adequacy conditions that we have established for it -- it was demonstrated that it can completely match and outperform the inductive capabilities of neural networks in the task of learning the past tenses of English verbs. This fact in and of itself does not entirely undermine the eliminative connectionist arguments based on the performance results of Rumelhart and McClelland's and MacWhinney and Leinbach's models. It remains to be shown that unlike neural networks, the SPA is able to represent explicitly the knowledge acquired in the learning process in the form of rules.

As we saw, the decision trees produced during the learning process are a useful symbolic format that lends itself to an easy conversion into production rules. This is done first by automatically pruning the tree of non-informative branches, i.e. branches that are either empty or classify very few exceptional cases. (Therefore, the rules shown below do not account for most irregular verbs.) Then the

value of each terminal leaf is selected as the consequent of a single production rule and the values of the attributes on all branches starting from the root and leading to this leaf are conjoined together as the body or antecedent of the rule. In this particular experiment, the production rules resulting from the conversion of almost all trees can be classified into three different groups, corresponding to the three different types of branches of a "typical" tree in Figure 3.4 -- identity mapping rules, suffix rules, exceptional changes rules. The identity rules were most numerous reflecting the fact that most of the transformations from input phonemes to output phonemes were identity transformations. Here are some randomly selected identity rules:

Rules derived from tree #1 (the rules produce the value of the first output attribute):

If I_1 (input attribute #1) = p,

then $\Omega 1$ (output attribute #1) =p

If I_1 = d, then Ω_1 = d

If I_1 = n, then Ω_1 = n

If I_1 = v, then Ω_1 = v

If I_1 = h, then Ω_1 = h

...

Rules derived from tree #4 (the rules produce the value of fourth output attribute):

If I_4 = i, then Ω_4 = i

If I_4 = p, then Ω_4 = p

If I_4 = m, then Ω_4 = m

If I_4 = s, then Ω_4 = s

...

The picture is essentially similar for all other trees -- they all produce a great number of identity rules. If we look at the tree structure of the first three trees, we can see that they are almost flat, producing mostly identity rules. This is due to the fact that only a very small number of verbs actually change in the first 1-3 phonemes. As we reach the fourth tree and beyond things begin to look different. There is already a sufficient number of verbs which end, and therefore have to receive a suffix in the past tense form. So the trees begin to branch out producing respectively identity rules (for the unchanging phonemes) and suffix-adding rules for the verbs that are three phonemes long. For example, some of these suffix-adding rules look like this:

Rules derived from tree #4 (the rules produce the value of the fourth output attribute):

If I_3 = k and I_4 = \$, then Ω_4 = t

If I_3 = p and I_4 = \$, then Ω_4 = t

If I_3 = l and I_4 = \$, then Ω_4 = d

If I_3 = r and I_4 = \$, then Ω_4 = d

...

What these rules tell us is the following: look at the third phoneme of an input verb form and look at the fourth, if the third is /k/ or /p/ and the fourth is blank, i.e. if

we are at a verb ending, then add the unvoiced /t/ as the fourth phoneme to the output form of the verb. Respectively, if the third input phoneme is /l/ or /r/ and the fourth is blank /$/, then add the voiced /d/ to the output form. Since all trees that map input patterns onto the middle portion of the output patterns have a very similar branch structure, it is not surprising to find that they all produce very similar production rules. For example, if we follow the productions for the same phonemes /l/, /r/, /k/, /p/ produced by the fifth, sixth, and seventh tree we will find the following rules:

Rules derived from tree #5 (the rules produce the values of the fifth output attribute):

If i_4 = k and I_5 = $, then Ω_5 = t

If I_4 = p and I_5 = $, then Ω_5 = t

If I_4 = l and I_5 = $, then Ω_5 = d

If I_4 = r and I_5 = $, then Ω_5 = d

...

Rules derived from tree #6 (values of the sixth output attribute):

If I_5 = k and I_6 = $, then Ω_6 = t

If I_5 = p and I_6 = $, then Ω_6 = t

If I_5 = l and I_6 = $, then Ω_6 = d

If I_5 = r and I_6 = $, then Ω_6 = d

...

Rules derived from tree #7 (values of the seventh output attribute):

If I_6 = k and I_7 = \$, then O_7 = t

If I_6 = p and I_7 = \$, then O_7 = t

If I_6 = l and I_7 = \$, then O_7 = d

If I_6 = r and I_7 = \$, then O_7 = d

...

These rules demonstrate that the SPA has been able to distinguish between verbs that receive a voiced or unvoiced suffix '-ed' in the regular past tense formation. But is the SPA able to express the knowledge it has acquired at a higher level? It is tempting to say yes but such an answer would be premature. What we have at this stage are rules that apply over positioned phonemes, i.e. these rules do not apply over any phoneme irrespective of its place in a verb (phonetic pattern). There is another inductive step to be made from these phonetic/ position specific rules to purely phonetic rules and then to morphological and lexical rules that possibly control the generation of past tense forms in adult speakers. We have to be very careful not to fall into the trap that connectionist researchers have fallen into by imagining that the induction problem of the acquisition of the English past tense can be solved simply at the phoneme/ position level. Ling and Marinov do not claim that by producing rules that are similar to suffix-adding rules of English the SPA has completely solved this induction problem. What the SPA has

been able to achieve is the extraction of position relevant phonetic rules which can be unified at a later stage to position irrelevant phonetic rules. These rules can form the basis for the full inductive step in the acquisition of the English verb paradigm.

Let us be more specific. The SPA cannot generalize from the position relevant phonetic rules to position irrelevant ones, e.g. it cannot make the step from propositional phonetic rules to first-order phonetic rules. But because it is able to represent its quite limited knowledge in rule form, it is not difficult to see how this knowledge can be generalized to first-order form. There are a number of symbolic learning programs, like GOLEM (Muggleton and Feng, 1990) for example, that can take all production rules generated by the SPA in an appropriate format and automatically generalize them, so that the new rules can become position independent first order phonetic rules. Such learning programs usually utilize the least general generalization (LGG) algorithm (Plotkin, 1970). In order to produce position irrelevant rules GOLEM has to receive the individual rules derived by the SPA in a form in which the position indexes can be represented as terms. For example the "raw" SPA rule

If I_5 = k and I_6 = \$, then Ω_6 = t

needs to be changed into

If I(5) = k and I(S(5)) = \$, then Ω(S(5)) = t

where S(X) is the successor of X. Although the conversion of

the indexes into terms can be done entirely automatically, the procedure in this case is strictly application specific and hence of little explanatory value.

Once all of SPA's rules indexes are converted into terms, they can be generalized by using the LGG algorithm. For example, the rules from the fifth and from the sixth tree:

If I(4) = k and I(S(4)) = $, then Ω(S(4)) = t

If I(5) = k and I(S(5)) = $, then Ω(S(5)) = t

are sufficient to produce the first-order rule:

For all n, if I(n) = k and I(S(n)) = $,

then Ω(S(n)) = t

In the same way the rules for /p/, /l/, and /r/ are generalized to first-order form:

For all n, if I(n) = p and I(S(n)) = $,

then Ω(S(n)) = t

For all n, if I(n) = l and I(S(n)) = $,

then Ω(n) = d

For all n, if I(n) = r and I(S(n)) = $,

then Ω(S(n)) = d

where n is a phoneme position and S is the next successive phoneme position.

One feature of the LGG algorithm is that it needs to generalize on the "right" set of clauses, otherwise it tends to produce overgeneralized incorrect clauses. Such incorrect overgeneralizations, however, can be removed completely when

negative cases are presented by using Shapiro's backtracking algorithm (Shapiro, 1981).

What these more general rules essentially tell us is that if a (verb stem) phonetic pattern ends at /k/ or /p/ we have to add to it the unvoiced (suffix) /t/ (in order to form the past tense of the verb); if it ends at /l/ or /r/, we have to add /d/. Similar rules for other phonemes can be produced. Note, however, that at this inductive step we still do not have explicitly represented information about the categories that are conveniently put in brackets -- 'verb stem', 'suffix', and 'past tense'. The rules that we have induced by generalizing the position relevant phonetic rules are still purely phonetic. We can easily see that they will give the wrong answer if the phonemes over which they range are part of an adjective or noun, rather than a verb stem. Clearly, we need to integrate these phonetic rules with morphological and lexical information to produce the correct rules for the past-tense formation in English. Simply observing phonetic regularities is not enough to complete the full inductive step.

But since both Rumelhart and McClelland's and MacWhinney and Leinbach's models as well as the SPA have access only to information regarding phonetic patterns, they cannot be said to have completed the full inductive step of the learning of the past tenses of English verbs. This is the reason why the SPA, even though it achieved much higher learning success

rates than its connectionist competitors, still failed to achieve the full accuracy of adult speakers. The major difference between the SPA and the connectionist models, however, lies in the fact that the SPA provides the basis for the solution of this inductive problem -- production rules ranging over phonemes -- whereas the connectionist models do not even begin to address the full induction problem. At first glance this contrast between the SPA and the rival connectionist models may not appear to be a significant one, but if we take a more global approach and ask how purely phonetic information is integrated in the overall functioning of the linguistic perceptual/ production system, the difference between them is obvious. If purely phonetic information cannot be represented in a form that can be readily interfaced and integrated with lexical, morphological, and syntactic information that is available to other language processing subsystems, it is hard to imagine how adult linguistic competence is developed. The connectionist models consider the acquisition of the English past tense as a totally isolated affair and thus fail to explain how the knowledge acquired by the networks is interfaced and integrated with other types of linguistic knowledge. The SPA, on the contrary, can represent the acquired knowledge as sets of production rules that can easily be unified into more general first-order phonetic production rules. These phonetic rules, in turn, can be used by other subsystems to complete

the full inductive step in learning the past tense of English verbs. The result of this theoretical comparison between the connectionist models and the SPA are summarized in Table 3.7. (See Table 3.7.)

Table 3.7

Inductive Steps in the Learning of the Past Tenses of English Verbs

Inductive steps to full competence	R&M	M&L	SPA
1. Learning of regularities between phonetic patterns	Yes	Yes	Yes
2. Representing the acquired knowledge in propositional phonetic rules	No	No	Yes
3. Generalizing to first order phonetic rules	No	No	Yes*
4. Possibility of Integrating first order phonetic rules with lexical, morphological, and syntactic rules	No	No	Yes

* SPA and the LGG algorithm

Conclusion

To summarize, we have established the following:

First, the eliminative connectionists's claim that there are no symbolic models capable of matching or surpassing the performance of artificial neural networks in the learning of the past tenses of English verbs is simply false. The SPA significantly outperforms Rumelhart and McClelland's as well as MacWhinney and Leinbach's models on this task without the use of TRICS and parameter tuning.

Second, in pattern association tasks where each pattern consists of a finite list of features whose values are finite and discrete, there is absolutely no advantage in the use ANNs over the SPA in terms of accuracy of learning, inductive generalizations, and speed of learning.

Third, by contrast with ANNs which are unable to represent knowledge in explicit form, the SPA can produce explicit representations in the form of production rules. The ability to represent knowledge explicitly is indispensable in solving the full induction problem of learning the past tenses of the English verbs.

In the light of these findings, we may conclude that eliminative connectionists' vision of cognition as pattern association and pattern recognition without symbolic representation is deeply flawed. Pattern association as such need not imply rule-less or cue-based models of language acquisition, or of human learning in general. If there are any

regularities in the sets of patterns that the SPA can learn to associate, they will be extracted and represented in symbolic form, ready for further processing. In contrast, the rival connectionist models do not offer any form of knowledge representation and leave the further processing and integration of the acquired knowledge a complete mystery. Because of this the eliminativist models cannot explain how mature speakers acquire their knowledge of the language. I hope that the theoretical and experimental comparison between the SPA and the rival connectionist models is sufficient to show that the rule-based symbolic approach stands a far better chance of explaining language learning, language processing, and cognition, generally, than eliminative connectionism.

ENDNOTES

[1]I adopt the qualifier artificial following the majority of connectionist researchers who realize that neural networks are neural or brain-like in a very abstract sense and do not see any chance at present or in the near future to integrate connectionist theory with actual neurophysiological findings (cf. Smolensky, 1988).

[2]The XOR (Exclusive OR) problem is the following: ANN has to learn the mapping

1 1 --> 0

1 0 --> 1

0 1 --> 1

0 0 --> 0,

i.e. it has to be able to output {0} when the input is {11} or {00}, and it has to be able to output {1} when the input is {10}, or {01}. This mapping turns out to be unlearnable by the two layered perceptron (Minsky and Pappert, 1969). A multilayer ANN, however, using the error backpropagation algorithm can learn this mapping, thus solving the XOR problem. (For some simple multilayer networks it can be demonstrated that there exists a distribution of weights for the network that solves the XOR problem without the use of backpropagation.)

[3]Some philosophers of science like Feyerabend (1975) claim that all theory changes are radical but they are able to do so only at the expense of changing the meaning of theory change.

[4]These results are very poor at least with respect to results produced by the SPA. Rumelhart and McClelland seem to believe that anything over 50% is a success for their model because a classic developmental study (Berko, 1958) indicates that children in early grade-school years produce the correct past tenses of novel verbs only 51% of the time. But even if this were so, the conclusion one must draw is that Rumelhart and McClelland did not provide a model for the mature system. They do not explain how the adults bridge this 50% gap.

[5]Ling and Marinov carried out several experiments with the full data set of MacWhinney and Leinbach, i.e. the set containing past participle, present participle, third person singular as well as past tense forms. The SPA very easily separated the four different forms on the basis of the explicitly coded categorial information and then proceeded to look for different regularities. The overall error rate dropped to 14.61%. However, since they consider the coding of categorial information in pseudophonetic form as entirely psychologically unjustifiable, Ling and Marinov did not attach any importance to this result. Also, MacWhinney and Leinbach

do not report any results on the full data set so there is no basis for comparison. The result shows, however, the power and versatility of the SPA.

APPENDIX I

GLOSSARY OF ACRONYMS

ASSISTANT - A diagnostic learning system member of the TDIDT family of symbolic learning systems. See Cestnik, Kononenko, and Bratko (1987).

ANN - [A]rtificial [N]eural [N]etwork. This term covers all single-layer and multilayer connectionist networks. A connectionist network typically consists of units (neurons) and connections between them. The units can be binary or real valued and the connections can have real valued strengths or weights.

BP - The error [B]ack[P]ropagation algorithm. The BP algorithm is the major connectionist learning algorithm. See Rumelhart, Hinton, and Williams (1986).

CART - A learning system member of the TDIDT family of symbolic learning systems. See Breiman, Friedman, Olshen, and Stone (1984).

C4 - An improved version of ID3. Has features that allow it to handle noisy data. See Quinlan (1986b).

ID3 - The most widely known member of the TDIDT family of symbolic learning systems. It has achieved significant advances in solving 'real-world' learning tasks. See Quinlan (1986a).

LGG - The [L]east [G]eneral [G]eneralization algorithm. See Plotkin (1970).

PDP - [P]arallel [D]istributed [P]rocessing. A general term describing the type of processing taking place in ANNs.

SPA - [S]ymbolic [P]attern [A]ssociator. A symbolic learning system member of the TDIDT family specifically designed to learn to associate sets of arbitrary patterns. It can solve pattern association tasks similar to the ones typically solved by ANNs. See Ling and Marinov (1992).

TDIDT - [T]op-[D]own [I]nduction of [D]ecision [T]rees. A general algorithm for the induction of decision trees from a set of examples. Originates from the Concept Learning System (CLS) of Hunt, Marin, and Stone (1966).

TM - [T]uring [M]achine.

BIBLIOGRAPHY

Armstrong, D. (1968). A materialist Theory of the Mind. London: Routledge and K.Paul.

Atlas, L, Cole, R., Connor, J., El-Sharkawi, M., Marks, R., Muthasamy, Y., Barnard, E. (1990). Performance comparisons between backpropagation networks and classification trees on three real-world applications. In: D. Touretzky, (Ed.), Advances in Neural Information Processing Systems, vol. 2, (pp. 622-29) San Mateo, CA: Morgan Kaufmann Inc.

Ballard, D. and Hayes, G. (1984). Parallel logical inference. In: Proceedings of the Sixth Annual Conference of the Cognitive Science Society, Rochester, NY.

Berko, J. (1958). The child's learning of English morphology. Word, 14, 150-177.

Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. (1984). Classification and Regression Trees, Belmont: Wadsworth.

Cestnik, B., Kononenko, I., and Bratko, I. (1987). ASSISTANT 86: A knowledge elicitation tool for sophisticated users. In: I. Bratko, and N. Lavrac, (Eds.), Progress in Machine Learning, Wilmslow: Sigma Press.

Chomsky, N. (1959). Review of B. F. Skinner's Verbal Behaviour. Language, 35, 26-58.

Chomsky, N. (1957). Syntactic Structures. Gravenhage: Mouton.

Chomsky, N. and Miller, G. (1963). Introduction to the formal analysis of natural languages. In: D. Luce, R.Bush, and E. Galanter, (Eds.), Handbook of Mathematical Psychology, vol.2, (pp. 269-321). New York, NY: John Wiley.

Churchland, P.M. (1988). Matter and Consciousness. Revised edition. Cambridge, MA: MIT Press.

Churchland, P.M. (1989). A Neurocomputational Perspective. Cambridge, MA: MIT Press.

Crick, F. and Asanuma, C. (1986). Certain aspects of the anatomy of the cerebral cortex. In: D. Rumelhart, J. McLelland, and the PDP Research Group, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 2, (pp. 333-371) Cambridge, MA: MIT Press.

Dennett, D. (1991). Mother nature versus the walking encyclopedia: A western drama. In: W. Ramsey, S. Stich, & D. Rumelhart, (Eds.). Philosophy and Connectionist Theory (pp. 21-30). Hillsdale, NJ: Erlbaum.

Feyerabend, P. (1975). Against Method. London: Humanities Press.

Fisher, D. & McKusick, K. (1989). An empirical comparison of ID3 and back-propagation. Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (pp. 788-793). Detroit, MI.

Fodor, J. A., (1975). The Language of Thought, New York, NY: Crowell.

Fodor, J. A. and Pylyshyn, Z. (1988). Connectionism and cognitive architecture: A critical analysis. In: S. Pinker and J. Mehler, (Eds.), Connections and Symbols, (pp.3-71). Cambridge, MA: MIT Press.

Fodor, J. A., and McLaughlin (1990). Connectionism and the problem of systematicity. Cognition, vol.35, 183-204.

Garzon, M. and Franklin, S., Neural computability. In: Omdivar, O. (Ed.), Progress in Neural Networks, vol. 1, (pp. 128-144). Norwood, NJ: Ablex.

Giles, C., Sun, G., Chen, H., Lee, Y., and Chen, D. (1990). Higher order networks for recurrent and grammatical inference. In: Touretzky, (Ed.), Advances in Neural Information Processing Systems, vol. 2, (pp. 380-397). San Mateo, CA: Morgan Kaufmann.

Goles, E. and Martinez, S. (1990). Neural and Automata Networks. Dordrecht: Kluwer.

Hartley, R. and Szu, H. (1987). A comparison of the computational power of neural network models. In: Proceedings of IEEE First International Conference on Neural Networks, vol.3, 17-22.

Hopcroft, J., and Ullman, J. (1979). Introduction to Automata Theory, Languages, and Computation, Reading, MA: Addison-Wesley.

Hunt, E., Marin, J., & Stone, P. (1966). Experiments in Induction. New York, NY: Academic Press.

Kononenko, I., Bratko, I., and Roskar, E. (1984). Experiments

in automatic learning of medical diagnostic rules. Technical Report, Jozef Stefan Institute, Ljubljana, Slovenia.

Lachter, J. & Bever, T. (1988). The relation between linguistic structure and associative theories of language learning -- A constructive critique of some connectionist learning models. In: S. Pinker & J. Mehler, (Eds.). Connections and Symbols (pp. 195-247). Cambridge, MA: MIT Press.

Ling, C. & Marinov, M. (1992). Answering the connectionist challenge: A symbolic model of learning the past tenses of English verbs, (forthcoming).

MacWhinney, B. (1990). The CHILDES Project: Tools for Analyzing Talk. Hillsdale, NJ: Erlbaum.

MacWhinney, B. & Leinbach, J. (1991). Implementations are not conceptualizations: Revising the verb model. Cognition, 40, 121-157.

Marinov, M. (1992). On the spuriousness of the symbolic/ subsymbolic distinction. Minds and Machines, (forthcoming).

McClelland, J., Rumelhart, D., & Hinton, G. (1986). The appeal of parallel distributed processing. In: Rumelhart, D., McClelland, J. & the PDP Research Group (Eds.). Parallel Distributed Processing, Vol. 1 (pp. 3-44). Cambridge, MA: MIT Press.

Minsky M., & Pappert, S. (1969). Perceptrons. Cambridge, MA:

MIT Press.

Muggleton, S. & Feng, C. (1990). Efficient induction of logic programs. Proceedings Of the First International Conference on Algorithmic Learning Theory, Tokyo: OHMSHA.

Newell, A., (1980). Physical symbol systems. Cognitive Science, vol. 4, 135-183.

Newell, A., Rosenbloom, P., and Laird, J. (1989). Symbolic architectures for cognition. In: Posner, I., (Ed.), Foundations of Cognitive Science, (pp. 93-132). Cambridge, MA: MIT Press.

Pinker, S. & Prince, A. (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. In: S. Pinker & J. Mehler, (Eds.). Connections and Symbols (pp. 73-193). Cambridge, MA : MIT Press.

Plotkin, G. (1970). A note on inductive generalizations. In: B. Meltzer & D. Michie, (Eds.). Machine Intelligence, Vol. 5, (pp.153-63). New York: North Holland.

Plunkett, K. & Marchman, V. (1991). U-shaped learning and frequency effects in a multilayered perceptron: Implications for child language acquisition. Cognition, 38, 43-102.

Pollack, J. (1991). Recursive distributed representations. In: Hinton, G., (Ed.), Connectionist Symbol Processing, (pp. 77-106). MIT Press, Cambridge, MA.

Putnam, H. (1960). Minds and machines. In: Hook, S. (Ed.).

Dimensions of Mind. New York, NY: New York University Press.

Pylyshyn, Z. (1984). Computation and Cognition: Toward a Foundation for Cognitive Science, Cambridge, MA: MIT Press.

Quinlan, R. (1986a). Induction of Decision Trees. Machine Learning, Vol. 1, 81-106.

Quinlan, R. (1986b). Probabilistic decision trees. In: Kodratoff,Y. and Michalski, R., (Eds.), Machine Learning: An Artificial Intelligence Approach, vol. 3, (pp. 140-52). San Mateo, CA: Morgan Kaufmann, Inc.

Quinlan, R. (1989). Unknown attribute values in induction. In: B. Spatz, (Ed.), Proceedings of the Sixth International Workshop on Machine Learning, (pp.164-68). San Mateo, CA: Morgan Kaufmann Publishers, Inc.

Quinlan, R., Compton, P., Horn, K., and Lazarus, L. (1987). Inductive knowledge acquisition: A case study. In: R.Quinlan, (Ed.), Applications of Expert Systems, Sydney: Turing Institute Press in assoc. with Addison-Wesley Publishing Co.

Ramsey, W., Stich, S., & Garon, J. (1991). Connectionism, eliminativism, and the future of folk psychology. In: Ramsey, W., Stich, S., and Rumelhart, D. (Eds.) Philosophy and Connectionist Theory, (pp. 199-228). Hillsdale, NJ: Erlbaum.

Rosenblatt, F. (1962). Principles of Neurodynamics. New York,

NY: Spartan.

Rumelhart, D. and McClelland, J. (1986). PDP models and general issues in cognitive science. In: D. Rumelhart, J. McLelland, and the PDP Research Group, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1, (pp. 110-146). Cambridge, MA : MIT Press.

Rumelhart, D. & McClelland, J. (1986). On learning the past tenses of English verbs. In: Rumelhart, D., McClelland, J. & the PDP Research Group (Eds.). Parallel Distributed Processing, Vol. 2 (pp. 216-271). Cambridge, MA: MIT Press.

Rumelhart, D. & McClelland, J. (1987). Learning the past tenses of English verbs: Implicit rules or parallel distributed processing? In: B. MacWhinney (Ed.). Mechanisms of Language Acquisition. Hillsdale, NJ: Erlbaum.

Rumelhart, D., McLelland, J. and the PDP Research Group (1986). Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1 & 2, Cambridge, MA : MIT Press.

Rumelhart, D., Hinton, G., and Williams, R. (1986). Learning internal representations by error propagation. In: D. Rumelhart, J. McLelland, and the PDP Research Group, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1, (pp. 318-362).

Cambridge, MA : MIT Press.

Schlimmer, J. (1987). Vote [Machine-readable data file], Adapted from Congressional Quarterly Almanac, 98th Congress, 2nd session 1984, Volume XL: Congressional Quarterly Inc., Washington, D.C., 1985.

Sejnowski, T. & Rosenberg, C. (1987). Parallel networks that learn to pronounce English text. Complex Systems, 1, 145-168.

Shapiro, E. (1981). Inductive inference of theories from facts. Tech. Rep. No. 192, Dept. of Computer Science, Yale University.

Shavlik, J., Mooney, R., and Towell, G. (1991). Symbolic and neural learning algorithms: An experimental comparison. Machine Learning, vol. 6, 111-143.

Siegelman, H. and Sontag, E. (1991). On the Computational Power of Neural Nets, Technical Report SYCON-91-11, Rutgers University.

Skinner, B. (1953). Science and Human Behavior. New York, NY: Macmillan.

Skinner, B. (1957). Verbal Behavior. New York, NY: Appleton-Crofts.

Smolensky, P. (1987). The constituent structure of connectionist mental states: A reply to Fodor and Pylyshyn. The Southern Journal of Philosophy, vol. 26, 137-161.

Smolensky, P. (1988). On the proper treatment of

connectionism. Behavioral and Brain Sciences, 11, 1-74.

Smolensky, P. (1991) Tensor product variable binding and the representation of symbolic structures in connectionist systems. In: Hinton, G., (Ed.), Connectionist Symbol Processing, (pp. 159-216). Cambridge, MA: MIT Press.

Smolensky, P. (1991a). Connectionism, constituency, and the language of thought. In: B. Loewer & G. Ray, (Eds.). Meaning and Mind: Fodor and His Critics (pp.201-228). Cambridge, MA: Basil Blackwell.

Stich, S. (1983). From Folk Psychology to Cognitive Sceince. Cambridge, MA: MIT Press.

Sun, G., Chen, H., Lee, Y., and Giles, C. (1991). Turing equivalence of neural networks with second order connection weights. In: International Joint Conference on Neural Nets, Seattle, vol. 2, 357-367.

Touretzky, D. (1991). BoltzCONS: Dynamic symbol structures in a connectionist network. In: Hinton, G., (Ed.), Connectionist Symbol Processing, (pp. 5-46). Cambridge, MA: MIT Press.

Valiant, L. (1984). A theory of the learnable. Communications of ACM, 27, 1134-1142.

Wickelgren, W. (1969). Context sensitive coding, associative memory, and serial order in (speech) behaviour. Psychological Review, 76, 1-15.

Weiss, S. and Kapouleas, I. (1989). An empirical comparison of pattern recognition, neural nets, and machine learning

classification methods. Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, pp. 688-693. Detroit: MI.

Wirth, J. & Catlett, J. (1988). Experiments on the costs and benefits of windowing in ID3. In: J. Laird, (Ed.), Proceedings of the Fifth International Conference on Machine Learning, (pp.87-99), San Mateo, CA: Morgan Kaufmann Publishers, Inc.