INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

ON THE INTERPRETATION OF NEURAL NETWORK ACTIVITY IN PARALLEL DISTRIBUTED PROCESSING MODELS OF COGNITION

M. A. Thesis presented
to the School of Graduate Studies and Research

by

S. Gregg Dahl

Department of Philosophy
University of Ottawa
December 1994

To Daedalus,

for his patience and appetite.

Introduction

Sometimes it takes years of debate for philosophers to discover what it is they disagree about. Sometimes they talk past each other in long series of books, articles, never guessing at the root disagreement that divides them. But occasionally a day comes when something happens to coax the cat out of the bag.

Daniel C. Dennett

"Evolution, Error, and Intentionality"
The Intentional Stance, pg. 287

Minds are one of the more interesting phenomena in nature. I regard the mind as a conceptual vista capable of being specified in many different and interesting ways. But in this work I am focusing on only those ways of describing the mind which aim to be the theoretical foundations of a science of the mind.

All generally philosophically acceptable conceptual formulations of a psychology worthy of being called a science stem from some form of "cognitivism" which explains psychological phenomena by appealing to internal mental processes and states. To date, the only form of cognitivism is "computationalism". Cognitive science has been methodologically dominated by "classical computationalism". Classical computationalism uses "programs" to provide the theoretical level of explanation for psychological capacities; the idea is that to have a certain psychological capacity is to have the capacity to execute a particular program. However, a relatively recent challenge to the classical computationalist methodology

comes from "connectionism". Connectionist researchers train artificial neural networks to perform particular cognitive tasks. Neural nets do not employ programs. This fact means that the same level of theoretic explanation of cognitive capacities enjoyed be classical computationalists is not directly available to connectionist theorists. I shall be examining the style of explanation a classical computationalist enjoys, some of the problems and objections made to computationalism, and finally, the possibility of a classical computational style of explanation being available to connectionist theorists.

I shall be following Robert Cummin's argument that psychological phenomena are not explained by theories which subsume such phenomena under causal laws, but are explained by viewing such phenomena as manifestations of capacities that are explained by analysis. The characteristic goal of explanations of natural phenomena via subsumption under causal laws is the formulation of, what Cummins calls, "transition theories". Such theories explain changes in a system from one state to another as cause and effect changes. So, transition theories explain events. But, Cummins argues that cognitive capacities are not events; they are dispositions, and dispositions are explained by what he calls "property theories". The goal of a property theory is to explain the properties of s system form the point of view of determining an answer to the question: "By virtue of what, does a particular system have a particular property?". A good property theory explains the instantiation of a property by a system. The strategy of a theorist seeking to explain a cognitive capacity is to analyze that capacity "functionally". A functional analysis decomposes the target "... disposition into a number of less problematic dispositions such that programmed

manifestation of these analyzing dispositions amount to a manifestation of the analyzed disposition". (Cummins, 1983, 28). A complete property theory for a cognitive capacity also demonstrates how the analyzing capacities are physically instantiated.

The other element of the cognitive science literature which will guide my work is David Marr's distinction of theories in cognitive science into Type-1 and Type-2. A Type-1 theory, according to Marr, adequately describes the decomposition of a cognitive capacity into its functional subcapacities such that the theorist knows exactly what is being computed and why. Type-2 theories describe the processes that solve a particular problem but do not have an adequately worked out computational description of the most abstract aspects of the problem they address. Type-2 theories can acquire their relevance by attaching great importance to the fact that a particular program solves a particular problem. Marr recommends that cognitive science can only claim to have achieved its goals if it has a Type-1 theory for a cognitive capacity. Type-2 theories do not satisfy the objectives of cognitive science.

The methodology of connectionism creates a rather unfortunate possibility for the connectionist approach to cognitive science. It may be the case that connectionist research will only yield Type-2 theories. The ability to produce Type-1 theories using the connectionist methodology is dependent upon the ability of connectionist researchers to give a principled semantic interpretation of the activity within neural

networks. This ability is directly associated with a connectionist researchers' ability to understand what is being represented within neural network activity and how it is being represented.

One of the main differences in the two methodologies employed in cognitive science is a commitment to, what Andy Clark (1989, 17) calls, "semantically transparent" systems. If a system is defined according to the computational rules it employs, then the symbols over which the rules operate may be interpretable as the elements contained in the purely conceptual analysis of the task the system performs. If those abstract entities which constitute the conceptual analysis of the task have discrete syntactic analogues in the computational specification of the system designed to perform the task, then the system is "semantically transparent". That is, both the physical tokens of the system's semantically interpreted symbols and the abstract primitive elements of the conceptual analysis of the task are governed by the formal rules of the system. Hence, what is for that system to have the capacity it does is the fact that it executes a particular set of formal rules. Classical computationalism is committed to designing semantically transparent systems. On the other hand, connectionism does not share this commitment. The reason is that the primary representational vehicle for a connectionist system is not symbols. The ramifications of this divergence in representational economies threatens to undermine connectionism's ability to adequately explain cognitive phenomena unless a semantic interpretation of the connectionist-style representations can be formulated.

Chapter one provides a detailed over-view of classical computationalism and its explanatory strategy. The focus is one the representation of knowledge in classical computational systems and the interpretive step required to yield theories of cognitive capacities. John Searle's objection to explaining the mind as the brain's program is introduced as "Searle's challenge". I argue that this challenge is not met by classical computationalism. I bring back Searle's challenge in chapter two which covers the essential characteristics of connectionist research in cognitive science and argue that his challenge produces a dilemma for connectionist theorists. The focus is on the suggestions made by theorists for semantically interpreting hidden-unit activity in neural networks. My conclusion addresses the threat connectionism faces of being explanatorily bankrupt in the face of producing only Type-2 theories. The difference in the representational economy of connectionism produces a different sort of theory. Theories which may not at present be adequately characterized.

Chapter One

All understanding is interpretation, and all interpretation takes place in the medium of a language that allows the object to come into words and yet is at the same time the interpreter's own language.

Hans-Georg Gadamer, *Truth and Method*
Second Revised Edition, pg. 389

Now as Uncle Hegel used to enjoy pointing out, the trouble with perspectives is that they are, by definition, partial points of view; the Real problems are appreciated only when, in the course of the development of the World Spirit, the limits of perspective come to be transcended.

Jerry Fodor, *Psychosemantics*
pg. 154

1.1 Classical Computationalism, Syntax, and Symbols

The stereo-typical classical computationalist conceives of the mind as a syntactic engine chugging along processing one symbolic state into another by strictly following a set of rules. When the rest of us think about our minds and then compare this introspection with the scientific accounts of mind, the rule-governed process suggested in the notion of a syntactic engine is not readily apparent. What is revealed by introspection is usually a panoply of beliefs, desires, wants, goals. and sometimes other "notions" which we may not be able to readily identify. But the scientific account of the mind called "computationalism" does not claim that minds, as thinking

things, can always (or, perhaps ever) truly introspect the nature of the workings of a syntactic engine; rather, computationalism claims our introspections are somehow related to the output stage of the processing of symbols. The remainder of this chapter will be dedicated to an outline of the classical approach to the computational approach to understanding the human mind.

To understand classical computationalism we need to begin by thoroughly examining the notion of a syntactic engine. Newell and Simon (1976) layout the fundamentals of a syntactic engine as a "physical-symbol system". They propose the following hypothesis as the guiding qualitative principle for the scientific investigation required to develop a general theory of intelligence: "A physical-symbol system has the necessary and sufficient means for general intelligent action." (179). Their hypothesis is intended to define the limits of a general class of systems the members of which behave intelligently. Human minds, syntactic engines, intelligent thinking things, etc., are all members of this general class of physical-symbol systems. Human minds are conceptualised as having a structure which is related to the structure of a physical-symbol system. To understand a physical-symbol system we need to look at the notion of a symbol as understood within the classical computational paradigm. Within this perspective, symbols are considered to lay at the root of intelligent cognitive behaviour.

Symbols are physical patterns which can combine to form more complex symbol-structures, which Newell and Simon (1976, pg 180) call "expressions". The combinatory power of symbols is determined by their formal syntactic properties, not

by their physical properties. Also, a symbol system is defined according to a collection of processes which are themselves specified by certain expressions. The processes which a particular system can perform are determined by the expressions which it can support. Thus, a physical-symbol system is a machine which manipulates, alters, creates, destroys, and performs in accordance to symbol structures. It is also an instantiation of a universal machine as conceived by Turing (1957).

The relationship between the common-sense world and a physical-symbol system is of primary interest when investigating the nature of intelligence. Newell and Simon (1976, 107) again embody a classical computationalist view on the nature of the above relationship with the following statement: " ... all information is processed by computers in the service of ends, and we measure intelligence of a system by its ability to achieve stated ends in the face of variations, difficulties, and complexities posed by the task environment." Computers can be programmed to achieve certain ends and so can be used to empirically investigate intelligence. But how does one computationally depict the task environment? This question is difficult to answer. The problem is that the "world" can be understood within a plethora of conceptual schemes. The computationalist construes the world as the source of input for all information-processing tasks facing an intelligent system. Specifying the input is, in some cases, difficult. But we do have a taxonomy of the possible states of an intelligent system interacting with the world. We call it Folk Psychology. In order to be intelligent, an information-processing system must "know, want, believe, desire, etc." certain things about the world or task environment. In other words, the system

behaves in such a manner that our folk psychological generalizations aptly depict its behaviour. But to achieve an explanation of intelligence we need more than a pragmatically adequate vocabulary. Pylyshyn (1984, 12) says: "We also want to be able to relate these generalizations to possible physically realizable mechanisms, and that's where computation can serve the important bridging role".

So, the relationship we are interested in when investigating intelligence can be described computationally. The world and the intelligent system coping within the world are conceptually related via computation. To be more specific, the two are related via a level of symbol-processing and a level of representation or semantic interpretation of the symbols. The world is represented within a physical-symbol system. The semantic interpretation of physical-symbol systems is our focus. But we should first take a look at the primary philosophical thesis required in order to conceptually connect symbols and our folk psychological generalizations.

1.2 Functionalism

Andy Clark calls functionalism the "natural bedfellow" of the computationalist committed to the physical-symbol system hypothesis (1989, 21). The functionalist

thesis originated in the work of Hilary Putnam (1960, 1975). Putnam claimed that as far as psychological generalizations are concerned, physical composition is not necessarily relevant. It does not matter how the thing to which the psychological generalizations apply is physically constituted.

Putnam's thesis created a new sort of materialism which sought to avoid the mystery of substance dualism and the troublesome identification of mental states with brain states. This new materialism is commonly known as "token-physicalism". Rather than types of mental states being identical to types of physical states, tokens of mental states are understood to be token identical to physical states. The relationship is nicely portrayed by Fodor (1987, 135): "According to standard formulations, to believe that P is to bear a certain relation to a token of a symbol which means that P. (It is generally assumed that tokens of the symbols in question are neural objects, ...)."

According to functionalism, the specification of a mental state is not related to the physical properties of that mental state. The specification of a mental state is determined by the abstract causal roles the mental state plays in its relationship to other mental states and in its relationship between the system in which it resides and that system's environment. That is to say, a mental state functions as a mediator between environmental inputs and behavioural outputs. A mental state also functions internally, affecting other mental states. Both abstract roles contribute to the individuation of types of mental states. According to Putnam (1960, 373) : "The

analogy which has been presented between logical states of a Turing machine and mental states of a human being, on the one hand, and structural states of a Turing machine and physical states of a human being, on the other, is one that I find very suggestive."

The physical structure of a Turing machine does not contribute to the definition of the machine, although any realization of a Turing-machine must be a physical object. A Turing machine is defined by its input-output profile and the specification of its internal state transitions: its "program". Thus, the move which Putnam finds suggestive is to consider characterizing the mental states involved in performing a particular cognitive task according to the description of the Turing machine capable of performing the same task. By doing so, cognition is viewed as computation. The program which defines the Turing machine capable of performing a particular cognitive task can be executed in different physical substrates. And so, cognitive tasks can be viewed as executions of programs. Such a picture of the nature of intelligence obviously sits well with the idea of using computers to empirically investigate mental phenomena. Functionalism denies the importance of the physical instantiation of the system capable of executing a program.

But there are is a problem with functionalism. Some systems behave in accordance with the functional specifications of mental states and yet our intuitions tell us to deny that such a system could possibly possess the mental state specified.

John Searle (1980) draws out these intuitions in his thought-experiment, "The Chinese Room". For Searle, genuine understanding is intrinsically related to the specific sort of physical substrate of our mental life. Accordingly, Searle denies the functionalist claim that physical instantiation of mental states is not relevant to being a genuine mental state. His argument is the following: "..., whatever purely formal principles you put into the computer, they will not be sufficient for understanding, since a human will be able to follow the formal principles without understanding anything." (Searle, 1980, 418). The person in the Chinese Room does not "understand" Chinese. Despite appearances to the contrary, he does not understand Chinese, he is simply following instructions for manipulating Chinese symbols; instructions that are written in English. The mental events which comprise the following of instructions are not those events which comprise the true understanding of Chinese. The person understands the English instructions, but Searle contends that the understanding of those instructions is due to the fact that the person knows what the English instructions "mean".

Searle takes issue with the fundamental premise of functionalism that the essence of the mental resides in the abstract causal roles a mental state plays. He argues that only the shadows of cognition are specifiable formally as a sequence of symbol manipulations. But to claim that the shadows are the real thing is a false claim, since, "... as far as we know it is because I am [or humans are] a certain sort of organism with a certain biological (i.e. chemical and physical) structure, and this structure, under certain conditions, is causally capable of producing perception, action, understanding, learning, and other intentional phenomena." (Searle, 1980, 421). The

formal properties of a cognitive task, as depicted by a program, are not the sort of properties which cause the system executing that program to have the ability to perform that task. Searle admits he may instantiate all sorts of programs, but that fact in no way explains the possession of the capacities depicted by those programs. Because such formal symbol-manipulations have only a syntax and no semantics, and intentional states are defined in terms of their semantic content; programs are not related to a computer in a way similar to the way mental states are related to the operations of the brain. The semantic interpretation of physical symbol systems ascribes content to those symbols via the cognitive capabilities of the interpreter, not because of the intrinsic nature of the manipulations depicted by a program. A program is not the product of a computer the way mental states and processes are products of the brain. The characterization of mental states and processes involved in a particular cognitive task as Turing-machine states and processes, which Putnam found theoretically interesting, is mistaken, according to Searle.

The mistake which results in the classical computationalist taking the formal shadows of cognition as the real thing can be traced to the notion of "information" at work in the view of intelligence as "information-processing". People process information in a sense which is far different from the sense in which physical-symbol systems process information. For a human, the symbols which constitute information have an intrinsic meaning which is not present to a physical-symbol system. Introspection can verify the presence of meaning. I can look at the sentence "The cat is on the mat" and know that the information such a sentence contains depicts a

situation where an actual cat is actually on an actual mat or it may depict a fictional situation. A computer does not "see" the relation between the information it is processing and the many possible worlds outside of it. That relation is brought into being by the interpreter of a physical-symbol system. So a computer does not do "information-processing" in the way a human does because the meaning of the symbols in a program is not information *for* the computer. To get the sense of information we need to have in order to understand that a computer is processing information in the way humans do, we must provide the meaning of what a computer is doing when it manipulates symbols. A computer has a syntax, but no semantics other than what we provide for it.

"Searle's challenge" must be answered. It draws into light the possibility that there is no relation between programs and minds in such a way that it challenges the defenders of computationalism to deny the importance of "meaning" for cognitive capacities. If only the abstract causal relations depicted by a program are the essence of the mental, then why interpret the elements of the program? Either meaning is important or it is not. Denying the importance of meaning would contradict the reason for semantically interpreting programs in the first place. I shall bring up Searle's challenge again when we examine the role of interpretation in connectionism. But for now, we need to carefully explore the details of the use of programs in the explanation of psychological capacities to make sure that Searle has properly portrayed the interpretive step required.

Functionalism is a very friendly thesis for the computationalist. Mental capacities can be explained by producing the proper program for a computer to follow. As long as the inputs and state transitions described by the program produce the appropriate output, and there exists a physical system capable of executing that program, then the behaviour of all physical instantiations of the program is explained as an execution of that program. The goal is to determine the proper functions that the programs compute, instantiate them in a physical system (like a computer), and then use the program as the theoretic explanation of the psychological task simulated by the computer. The last step in this process is the level at which interpretation plays a key role (and a flawed role according to Searle).

1.3 Programs and Psychological Explanations

Before examining the role of interpretation in the computationalist explanations of cognitive capacities, I want to point out the worries expressed by some of the leading theorists in the field. The specification of the proper cognitive function involved in a particular cognitive capacity has proven to be a more than formidable task in many instances. David Marr (1977) and others (Clark (1989), Cummins and Swartz (1988)), all suggest that for some cognitive tasks, no proper function exists. But suppose that a particular cognitive task is completely specified by some computable function. Now, what does this assumption bring with it in terms of

required conceptual manouvers? According to classical computationalism, the arguments and values of a cognitive function consist of the concepts of folk psychology. The initial analysis of a cognitive task serves to discern the contributing elements in accordance with the taxonomy of folk psychology. The most important element is to determine what a system needs "to know" in order to have the capacity for certain tasks. This process is the hallmark of what Cummins (1979) calls "functional analysis". The idea is to break down cognitive capacities into a number of less problematic capacities which can be programmed in such a way that there aggregation manifest the original analyzed capacity.

The major thesis Cummins defends in The Nature of Psychological Explanation says that psychological phenomena are explained by analysis rather than subsumption under causal laws. Analyzing the capacities which underlie psychological phenomena explains those phenomena, provided the analysis has certain characteristics. A "functional analysis" specifies the inputs, outputs, and internal state transitions of a system according to abstract causal roles. "Interpretive analysis" specifies the relevant inputs and outputs of a system via a semantic interpretation. Cummins suggests that it is "interpretive functional analysis" which explains the possession of psychological capacities. By analyzing a particular psychological capacity into an array of sub-capacities, whose physical instantiation is readily understood, the analyzed capacity is explained as a manifestation of the collective array of sub-capacities. Revealing the difference between subsumption and analysis will help to clarify the sort of psychological explanations Cummins is arguing for.

He begins by distinguishing between a "transition theory", which plays a central role in explanations involving subsumption, and a property theory, which is key to an analytic explanation. To clarify the difference between the two types of theories, Cummins (1979, 15) writes:

> The characteristic question answered by a transition theory is: Why does system S change from states s-1 to s-2 ? The characteristic question answered by a property theory is: What is it for system S to have property P ?

We want to have answers to questions such as, "What is it about humans that makes them intelligent?" or "What is about humans that allows them to solve certain problems?". So, a psychological theory having any explanatory power will begin with a property theory since the most important questions we want answered have the characteristics of those which are answered by property theories. Having a property theory for a specific psychological capacity involves performing a compositional analysis of S and a functional analysis of P. The property theory is explanatory when it is shown that a system having the sort of composition of S is bound to have property P. But a psychological theory must also give an account of how the analyzed capacity is physically instantiated by the system being studied.

According to Cummins, psychological capacities are inferentially characterizable dispositional capacities. To possess a particular psychological capacity

is to have a certain disposition which can be understood as a relation between certain behaviours given certain conditions. In terms of a functional analysis, the relation is between certain outputs by a system given a particular set of inputs to that system. The relation is described abstractly as an inference or computation. Some relations may be more complex than others, in which case a functional analysis of a relation will proceed until the most basic relations needed to explain any particular psychological capacity are demonstrated to be dispositions of the physical structure of the system possessing the analyzed capacity. At bottom, then, the most basic elements in an analysis of the physical structure of a system are isomorphic to the most basic elements of a functional analysis of that same system's psychological capacity. The physical structure of the system under study is interpreted as having the dispositions inherent to the functional analysis of system's psychological capacity.

Explaining a psychological capacity is, therefore, a two-step process. A theorist begins by performing an inferential analysis upon the psychological capacity she wishes to explain. As an example, suppose one wished to explain the capacity for being able to play euchre. Begin with the most obvious tools for analysis: a knowledge base comprised of a set of representations (both tacit and explicit) to represent each card in the deck used for playing and the rules for the game. Now, how does our researcher proceed? Much of playing euchre involves making decisions such as which suit, if any, to make trump, which card to lead a round with, etc., etc, ... , which are all complex decision-making capacities that when taken in aggregate manifest the capacity to play euchre. Obviously, we have here a capacity suited for

further inferential functional analysis. The analysis of each sub-capacity stops when the capacity needed can be mimicked by a physical process such as electrical current through a relay switch. I do not wish to create the impression that all capacities will break down in the way hoped for. Some capacities may rely upon inferences which may not be within the reach of deductive inferences. We need to assume at the outset that the capacities we want to analyze will break down into a set of deductive inferences. In our example of playing euchre, let us assume that the capacity for participating competently in a game is inferentially characterizable, and that our theorist has written the set of instructions, or "program", which is the computational description of the task.

The second step our researcher must take involves physically instantiating the abstract computational description of the task by having a computer execute the program. The machine itself is capable of executing (via a specification of a particular algorithm) the set of basic elements specified in the computational description of the task. The aggregation of these basic elements and operations, arranged in the order determined by the program, manifests the capacity to competently engage in playing a game of euchre.

At some level in the process of analyzing the capacity to play euchre (by assumption, an inferentially characterizable capacity or ICC), the structure of the inferential analysis resembles the structure of the physical description of the system possessing the capacity. As Cummins states (1978, 36):

> There must be, ... , an isomorphism of structure between the information-processing program and some program couched in descriptive terms known to be executed by the target system.

One program is written in terms used to describe symbol manipulations. The other program is comprised of a descriptive vocabulary for physical processes. The program dealing with symbols can be interpreted in any number of ways depending upon the semantic interpretation of its primitive elements. If the structures of the two programs are isomorphic, then the semantic interpretation of the symbol manipulation program can be applied to the physical process program. Our researcher thereby acquires an explanation of what it is for the physical system to have the capacity analyzed. The explanation is founded upon the semantic interpretation of the information-processing (or symbol-manipulation) program.

Cummins points out the worry expressed by critics of the computationalist approach to explaining cognition (1978, 37) :

> The problem is that, given any physical physical transaction you like and any symbolic operation you like, there will generally be some set of conditions -some context- in which that physical transaction would count as a performance of the symbolic operation.

Critics of computationalism point to the indeterminacy of interpretation of physical processes as the fatal flaw for the enterprise. Searle, shown above to be one of the

staunchest critics of computationalism, points at the computational possibilities for an ordinary wall as the reductio argument against the use of programs as explanations of cognitive capacities. Searle (1990) says that using the theoretical vocabulary of particle physics as the vocabulary for specifying the descriptive program of the physical processes in a wall, and then circumscribing a particular context in those wall processes, could in principle, lead one to the conclusion that the wall is performing the task specified by the information-processing program that is isomorphic to the wall's descriptive program. Searle cannot grant processes like "understanding" to any physical structure that is not a human brain-like structure.

The computationalists' response to the problem of indeterminacy of interpretation is to provide stringent conditions upon the type of isomorphism of structure required to exist between the descriptive program and the information-processing program. Zenon Pylyshyn (1984, 178) deals directly with the problem of constraints on isomorphic structures by examining the nature of physical-symbol systems at the level of transduction and sees "the need for a principled constraint on what can count as a transducer". This constraint is intended to give us a clearer understanding of the set of things which we consider to be doing the translation between the descriptive and the information-processing programs. The constraint is intended to limit the available possible interpretations of descriptive programs. Among the set of possible interpretations for the information-processing program is *the* correct interpretation. The structure of that interpretation is to be isomorphic to the structure of the descriptive program. Given this relation between the two

structures, how are we to understand the role of the transducers facilitating the translation between programs? A lengthy quote from Pylysyn (1984, 178) reveals the sort of role he envisions for the transducers he would count as actual transducers:

> Roughly speaking, a transducer produces a code, C, of bounded size -say, an n-tuple of symbols- whenever it is stimulated by an event that is a member of a class of events that can be given a finite physical description, D, cast in terms of the primitive vocabulary of physics (or chemistry or some other basic natural science). It is an approximation only to the extent that the relation between D and C can be modulated - primarily along the dimension of intensity - by the state of the organism. It cannot be freely altered to, for example, produce a code, C', which normally is produced by a different description D'. Specifically, it cannot be modified by the cognitive system so as to change the semantic content it imparts to symbols in the cognitive system, though it can be affected by certain adjustments to sense organs, for instance, by adjustments in sensitivity or even general movement, as in the case of changing the direction of gaze.

The constraint Pylyshyn wants to put on the relation between C and D is such that C and D must be isomorphic to the extent that the only difference allowed is along a dimension of intensity. This constraint also has a scale by which it is judged. The extremes of this scale are "strong" and "weak" equivalence. Weak equivalence amounts to a behavioural equivalence between the two systems specified by C and D. This sort of equivalence is pejoratively labelled as "mere mimicry". Strong equivalence amounts to the condition that both programs can be represented by the same program (say "P") in an information-processing program for some specified

virtual machine. Pylyshyn justifies the theoretical pursuit of P by suggesting that the search may reveal some interesting middle ground between the extremes and that some point within this middle ground may suffice as a constraint upon the relation between D and C (see also Scott and Strachy (1971)).

Pylyshyn's reasoning leads one directly to the worries as expressed by Searle in "Is the Brain a Digital Computer?" (1990) where he extends the notion of the observer relativity of the semantics of syntactic structures; not only is semantics not intrinsic to syntax, but syntax is not intrinsic to physics. The interpretation of physical processes as syntactical is observer relative. In the context of Pylyshyn's notion of strong equivalence, the equivalence is between an observer-relative computational interpretation of the physical processes and a description of those physical processes by a natural science. "As applied to the computational model generally, the characterization of a process as computational is a characterization of a physical system from outside; and the identification of the process as computational does not identify an intrinsic feature of the physics, it is essentially an observer relative characterization." (Searle, 1990, 28). But, how are we to make sense of the notion of an intrinsic feature of a physical system that is specifiable via some non-observer relative characterization? I am willing to admit that the computational characterization of a physical system is , as it were, from "the outside". But any characterization of a physical system is from "the outside". The point Searle makes does not apply to only syntactic computational characterizations, but to all characterizations of physical processes. The characterization of a physical process must always take place from

some perspective outside of that process. And as Searle admits, he cannot claim there are a priori limits on the characterization of patterns we can observe in nature. The strength of the notion that syntax is not intrinsic to physics is rather negligible if Searle's only point for formulating it was to imply that physical processes can only be characterized syntactically by an observer.

But Searle points out another aspect of the notion that syntax is not intrinsic to physics: syntax has no causal powers. Now the point about the characterization of physical processes as computational processes has more power; but only if the computational characterization of physical processes is understood to provide the foundation of a causal explanation for the psychological phenomena manifested in those same physical processes. Searle is quite correct when he discounts the use of computation in causal explanations of psychological phenomena:

> ... you cannot explain a physical system
> ... by identifying a pattern which it shares with its computational simulation, because the existence of the pattern does not explain how the system actually works *as a physical system.*
>
> ... the most we could find in the brain is a pattern of events which is formally similar to the implemented program in the mechanical computer, but that pattern, as such, has no causal powers to call its own and hence explains nothing. (1990, 32-33)

Searle's charge only works against those who see programs as causal explanations of psychological capacities. The isomorphism of structure between a physical description of a process and a computational program is not, to my knowledge, ever claimed to

apply to causal properties. The sort of isomorphism of structure invoked by Cummins is not an isomorphism of causal properties. If computationalists were purporting to be formulating transition theories on the basis of a computational characterization of physical processes , then Searle's charge would be deadly to the whole idea of explaining cognition computationally. But computationalists are careful not to make these sorts of claims, at least since the work of Robert Cummins. Thus. Searle's charge falls short of applying to Cummins' notion that psychological capacities are explained via property theories rather than transition theories. A program can be used in the formulation of property theories.

But Searle's challenge, as outlined above in the section on functionalism, remains in full force. The interpretive stage of the formulation of property theories for cognitive capacities is an observer relative interpretation. Programs acquire an explanatory dimension in the context of formulating property theories because they are descriptive, under a specific interpretation, of what happens within the system possessing the capacity to be explained; or in other words, what it is for the system to have that particular capacity. The computational description is in no way a causal account of the capacity being investigated. As Cummins (1977) is careful to point out: a program is a theorist's tool, not a cause. An information-processing program gives a cognitive theorist a tool for devising property theories of cognitive systems. The notion of a cause, says Cummins, is related to the idea of explaining natural phenomenon via subsumption under physical laws; an explanatory strategy which does not suit the domain of psychology.

But the sort of characterization of computational processes required to satisfy the demands of formulating a property theory for cognitive capacities is different from the characterization of physical processes as computational. The later characterization does not imbue physical processes with any meaning; it simply takes note of certain abstract properties which are common to the descriptions of a physical system cast at different levels of abstraction. On the other hand, the characterization of computational processes required to formulate property theories of psychological capacities does imbue meaning to computational processes and that difference is enough to keep Searle's challenge valid. He does not need to make the claim that syntax is not intrinsic to physics in order to have a valid objection to computationalism. Computationalism does not purport to give causal explanations of psychological capacities. Therefore, the observer relativity of interpreting physical processes computationally is not a threat to the goals of computationalism. But the observer relativity of semantically interpreting computational processes is a threat to the notion of information-processing at work in computationalism. One way of answering Searle's challenge may be to change the style of computational processes to non-syntactical. Semantics may not be intrinsic to syntax, as Searle says; but would Searle have to expand his claim to keep up his challenge in the face of connectionist-style computationalism? This non-syntactic style of computationalism must depart from the sort of interpretive stage required of programs in the formulation of property theories in order to have a chance of meeting Searle's challenge while at the same time not claiming to be a methodology for formulating causal explanations of cognitive capacities. Before getting to the consideration of such prospects for

connectionism, it will be helpful to have before us David Marr's description of the levels of explanation in computationalism.

1.4 Levels of Description in Computationalism

David Marr (1977) discerns three levels of theory within the computational methodology for investigating a particular cognitive capacity. Level-1 is the "computational theoretic" level in which the abstract structure of the task is specified. Marr (1977, 133) states: "One can think of this part as an abstract formulation of *what* is being computed and *why*, and I shall refer to it as the theory of a computation". The computational-theoretic level addresses the most abstract aspects of a solution to an information-processing problem. This level is "the search for P" as advocated by Pylyshyn. Level-2, in Marr's scheme, is the algorithmic level. Particular algorithms which implement a Level-1 theory are specified at this level. Of course, the fact that there could be many different algorithms for a Level-1 theory fuels the sorts of worries about indeterminacy of interpretation that Pylyshyn addressed. The final theoretic level in computationalism according to Marr is the physical instantiation and practical demonstration of the Level-2 algorithm. Level-3 is physical implementation level, or what most philosophers would call the level of realization. This level is the same level at which Newell and Simon aim their suggestion to use computers to empirically investigate intelligence.

Marr also specifies the types of theories in computationalism. Any information-processing problem which can be given a complete computational theory (i.e. all three levels addressed) has a "Type-1" theory. If the theorist knows exactly what information is being processed at Level-2, then she has a Type-1 theory of a particular cognitive capacity. A theorist has a Type-2 theory if she has an algorithm which solves a task problem but she does not have a computational theory adequately worked out for that task problem. She must have all the proper elements in place at Level-1 before she can consider herself as having a Type-1 theory for the capacity she has investigated.

Marr (1977, 134) uses the ability to pay chess to illustrate what he considers to be inadequate element within a Level-1 theory. The goal of chess is to "take the opponent's king". But having this sort of characterization within a Level-1 specification of the information-processing problem of playing chess will not suffice. The computational theory is intended to specify exactly what is going on. "Taking the opponent's king" may be suitable as a general characterization of the behaviour of a chess-playing system, but it does not indicate what computations are being employed. Despite the lack of a proper Level-1 theory, it is possible to construct an algorithm which, when executed, performs the task under study. Marr considers such accomplishments to be somewhat lacking in comparison to the Type-1 situation where all three levels of theory are complete.

The computational theory is the level of theory that is the foundation from

which an understanding of human cognitive capacities is formulated via semantic interpretation. Marr does not think that Type-2 theories enlighten us in relation to the puzzle of human cognition. However, Marr does suggest that a Type-2 theory can serve as the basis of inference for determining more than just a particular solution to a particular problem. He also suggests (1977, 143) that some problems may well not have a computational theory: "This can happen when a problem is solved by the simultaneous action of a considerable number of processes, whose interaction is it simplest description, ... ". What Marr is suggesting is that the complexity of the manifestation of a cognitive capacity may not allow a theorist to formulate a computational description of Level-2. The move to Level-2 from Level-1 involves a semantic interpretation which may not always be possible. I do not want to make too much of what Marr is suggesting, other than to point out that his suggestion supports the motivation for Connectionist-style computationalists to analyze their cognitive models at a level higher than their literal description (more on this aspect of computationalism in chapter two).

The task for a theorist is to establish a theory of a system's capacity which is more than a literal description of the activity within a system. To complete the task, activity within a system must be interpreted, as suggested by Cummins and Pylyshyn. The literal description mentioned by Marr is significantly similar to what Cummins calls the descriptive program of a targeted system. Pylyshyn would agree that a Type-2 theory is adequate, if all we want to be able to understand is physical-symbol systems which mimic human cognitive capacities. A Type-2 theory would give us the

situation Pylyshyn describes as "weak" equivalence. The goal is to strive towards "strong" equivalence by determining what Pylysyn calls "P", the information-processing description of the task for a virtual machine. Marr calls "P" the theory of a computation. Both Marr and Pylyshyn are in agreement over the importance of having a Type-1 theory for an information-processing problem, if what we are after is an understanding of intelligence, an understanding of mind. The crucial link between Level-1 and Level-2 is the semantic interpretation of the primitive elements constituting Level-1 (symbols). This interpretation is applied to all computational levels and is also applied to the physical descriptive program that is structurally isomorphic to the theory of computation. Before concluding our look at classical computationalism, we need to examine more closely the style of representation employed.

1.5 Representation in Classical Computationalism

The symbol manipulations specified in the theory of a computation can represent, in principle, any process in an equally specified domain. The representations used in a classical computational theory of cognition are symbols. Lars-Erik Janlert (1987) examines the metaphysics of the style of representation at work in classical computationalism.

Janlert begins by asking us to consider the world as a sequence of situations. A single situation is the complete state of affairs at a single moment in time. A complete state of affairs is specified by a set of laws of motion (including any constants and particular values for any variables in the set) which are in turn specified by sentences in first-order logic. Given a complete description of a situation, laws of motion, and an action (collectively, this description is a set of axioms); the situation resulting from an action can be deduced from the sentential specification of the action in relation to the other elements of the set of axioms. All changes induced by an action can be deduced. But all the relations between axioms which remain unchanged must also be deduced. The inferences associated with any change must all be computed even if a change has no relevance to any members of the axiom set which specifies a situation. The frame problem arises because of the inability of a deductive inferential system to "frame" adequately the relevant, and only the relevant, changes in its axiom-set. As Janlert describes it, when a deductive inferential system has the frame problem: "The outcome is a problem solver sitting in the midst of a sea of axioms representing non-changes, endlessly calculating non-effects." (1987, 6).

Janlert suggests that the concept of a situation (which collectively constitutes the "world" of a system) is an unsuitable foundation for a metaphysics of change. The main problem with the "situation" concept is an inability to make use of "situations" when defining other concepts which would be useful in common-sense situations: concepts such as action, event, change, oddity, etc. . Representations used by classical computationalists do not have the sort of flexibility we seem to require. An

important distinction between types of representations in classical computationalism is the explicit-implicit distinction. All axioms in the base axiom set are considered to be explicitly represented. The inferential entailments of those explicit representations are considered to be implicitly represented by the system. At Level-1, the theory of a computation, this conceptual distinction is useful form the point of deciding how and what is being computed, and specifying the parameters of the problem space. But at the algorithmic level, difficulties arise when trying to specify procedurally which information available (either explicitly or implicitly) in a problem space is to actually be used by a system which is able to competently manoeuvre within that same problem space. Janlert (1987, 35) notes the following reason for this difficulty:

> A given body of information can be axiomatized in many different ways, if it is at all axiomatizable. There is considerable freedom of choice in deciding what should be explicitly and what should be implicitly represented, what should be taken as primitive and what as non-primitive.

From this point, Janlert moves to talking about the metaphysics of representations in classical computationalism. He talks of "real" properties in the world. Real properties are then conceptually analyzed as primary and secondary. In the realm of real properties in the world, explicitness is a self-defeating quality because the more information available explicitly the more trouble a system has determining what not to consider. This problem cannot be ignored but Janlert suggests a way in which it may be possible to overcome it. The solution is to come up with a form of representation that could have a form sufficiently isomorphic to the form of the real

world properties; since we seem to be able to cope by employing real world properties in solutions to problems we face.

But, as an expression of caution toward the likelihood of ever realizing such a solution, Janlert gives an argument. The argument addresses the possibility of a "minimally explicit representation". The abstract space used to describe the collective organizational structure of this style of representation is a vector space. But vector spaces cannot possibly be good models of "real world" properties. According to Janlert,

> ... to ensure a manueverable model, it is also necessary that the choice of "base vectors" corresponds to the structure of the problem world in such a way that the relevant changes and actions can be easily analyzed into components that correspond to the base vectors spanning the description space. (1987, 36)

He adds an interesting footnote to this claim:

> Here, the vector space metaphor breaks down, since we cannot expect that the description space is anything like linear.

If the style of representation required to ensure the sort of flexibility we need to accommodate changes in the real world properties is limited to being described by only linear spaces, then Janlert is correct. Classical computationalism is limited to specifying its problem spaces using linear spaces. The non-linear nature of the relations between the real world properties of a problem space is incongruent with the linear nature of the computational description of the same problem space. At best, one

can only expect to achieve close approximations when using linear relations to describe non-linear spaces.

Classical computationalism employs representations which are too rigid. The description spaces of problems are confined to being linear spaces. As is indicated above, the problems which require flexible approximate solutions do not reside in spaces which can be as linear. The need for a minimal explicit representation which is less rigid cannot be handled by classical computationalism. But connectionism does hold out the possibility of being able to employ minimal explicit representations. The description spaces for problems faced by connectionist systems are not limited to being linear. The flexibility of connectionist representations is one of their redeeming qualities, but at a cost. The semantic interpretation of connectionist systems is more difficult. Connectionist representations can reside in non-linear description spaces. Specifying the content of a representation in the description of a connectionist problem space will be the focus of the chapter two.

1.6 Conclusion

I want to conclude our look at classical computationalism by drawing together the main characteristics of the style of inquiry into intelligence which has a methodology that is committed to constructing what Clark (1989) calls "semantically

transparent" formal systems. This characterization identifies systems in which the interpretation given to the constituent elements of the conceptual level of a problem is carried through the other levels which depict the same problem. The interpretation applies to representations of propositions encoded in language-like syntactic structures.

Any form of computationalism is committed to the assumption that intelligence has a computational specification. The physical realization of a computational specification is not relevant (or in terms of the functionalists, the computational specification has multiple physical realizations). These two characteristics are common to both classical computationalism and connectionism. The two characteristics that differentiate the two styles of computationalism are: 1) the physical symbol system hypothesis, and 2) the assumption that the world is comprised of discrete states of affairs (or "situations"). The last two characteristics commit the classical computationalist to a particular representational economy employing symbols which are supposed to represent the elements of the world the mind employs in order to cope within particular situations. But connectionism is not committed to the last two characteristics. The representational economy of a PDP system is different. Whether that difference is significant to the point of overcoming the problems faced by classical computationalists has yet to be determined. Connectionism must employ interpretation of is representational economy if it is to explain psychological phenomena by formulating property theories, and so Searle's challenge must be faced by the connectionists.

Chapter Two

Despite the great wealth of our languages, the thinker often finds himself at a loss for the expression which exactly fits his concept, and for want of which he is unable to be really intelligible to others or even to himself. To coin new words is to advance a claim to legislation in language that seldom succeeds; and before we have recourse to this desperate expedient it is advisable to look about in the dead and learned language, to see whether the concept and its appropriate expression are not already there provided. Even if the old-time usage of a term should have become somewhat

uncertain through the carelessness of those who introduced it, it is always better to hold fast to the meaning which distinctively belongs to it (even though it remains doubtful whether it was originally used in precisely this sense) than to defeat our purpose by making ourselves unintelligible.

Kant, *Critique of Pure Reason*
2nd Part, 2nd Division, Book 1 Section 1
(Norman Kemp Smith translation)

2.1 Connectionist Computationalism

If the general philosophical opinion can be trusted, Connectionism is the hottest prospect for advances in cognitive science research. The limitations encountered in research guided by the methodology of classical computationalism have dampened the

expectations for artificial intelligence that were voiced in the earliest days of the enterprise. Researchers now seem to be very careful when speculating on the prospect of non-biological intelligent beings sharing space with us on this planet. The focus for this chapter will be the features of the Connectionist (or Parallel Distributed Processing (PDP), neural net) model for investigating intelligence and its methodology for formulating scientific explanations of psychological capacities.

I shall begin my discussion of the PDP model of cognitive processes and states by examining an argument advanced against computationalism by Dreyfus and Dreyfus (1988). The philosophical heritage for their argument is located in the works of Husserl, Heidegger, and Wittgenstein. They conclude their paper with the following claim targeted against connectionism:

> If Heidegger and Wittgenstein are right human beings are much more holistic than neural nets. Intelligence has to be motivated by purposes in the organism and goals picked up by the organism from an ongoing culture. (1988, 331)

The support Heidegger and Wittgenstein lend to Dreyfus and Dreyfus' argument is drawn from the formers' notions concerning the non-formalizability of common-sense knowledge. Both Heidegger and Wittgenstein argued that common-sense understanding does not necessarily lend itself to being articulated in the way hoped for by rationalist thinkers. Dreyfus and Dreyfus point out that humans may not employ common-sense knowledge in the way "knowledge" is understood by the classical computationalist, i.e. a set of premises and deductive inferences. If humans

cope within a particular situation by means other than by making inferences; if they are able to cope simply because of the way they *are*, then the search for a theory of these skills using the classical computationalist methodology will be in vain.

The underlying assumption supporting Dreyfus and Dreyfus is that human capacities for intelligent behaviour do not reduce to a set of inferences appropriate to the situation in which intelligent behaviour occurs and that those inferences cannot be aggregated in a way which would resemble anything like the life of an intelligent entity. Intelligence cannot be analyzed the way classical computationalism wishes to analyze it. Dreyfus and Dreyfus align the classical computationalist approach to understanding intelligence with the rationalist conviction that every phenomena must have a theory to explain its nature. Humans display an ability for competently coping with demands placed upon them within certain situations. The fact that humans come up with solutions to such problems cannot be denied. What is denied by Dreyfus and Dreyfus is the rationalistic-style claim that a theory of this problem-solving ability is possible. They dismiss any approach to understanding this human ability which assumes that it involves inferences from a knowledge base. They ask rhetorically:

> ... is the common-sense background rather a combination of skills, practices, discriminations, and so on, which are not intentional states and so, a fortiori, do not have any representational content to be explicated in terms of elements and rules? (1988, 322)

While the arguments Dreyfus and Dreyfus advance against classical computationalism are effective; even though they do not constitute a knock-out punch, their effectiveness

against connectionism is somewhat mitigated by the fact that Dreyfus and Dreyfus admit that connectionism may eventually serve to give evidence for the truth of their claim that the domain of cognitive capacities does not, by its very nature, lend itself to a theoretic understanding. Theories are understood to articulate the relationships between objective elements according to abstract principles. The structure of the domain addressed by a theory is reflected in the structure of the theory of that domain. Dreyfus and Dreyfus content that the domain of cognitive capacities may not have a theoretical structure (1988, 326), which would account for the lack of success of classical computationalism and the eventual support for their position from connectionism. When a cognitive theorist tries to find the primitive elements which will constitute her theory, she must free them from their context. The downfall of cognitive theories, according to Dreyfus and Dreyfus, occurs at this point:

> ..., we are in effect trying to free aspects of our experience of just that pragmatic organization which makes it possible to use them intelligently in coping with everyday problems. (1988, 320)

The attempt to isolate and assemble into a theory the elements involved in intelligent behaviour destroys the essential characteristic of those elements that allows humans to behave intelligently. A theorist cannot capture the structure of the organization of the elements as they are actually used by humans because they do not have a structure which can be theoretically described.

The philosophical differences between critics such as the Dreyfuses and computationalists rest on this point about theoretical structures. The holism advocated

by one side of the debate dictates the impossibility of constructing a theory which could have a structure isomorphic to a computational theory. In terms used by Pylyshyn, Dreyfus and Dreyfus argue that the pursuit of "P" (or of strong equivalence) assumes what they deny. The structure of the cognitive domain is not such that it will ever be described by a program for some virtual machine. A program uses abstract principles to organize a set of context-free elements into certain relationships. To free these elements from their context is to free them from their usefulness. Or, more precisely, to try to interpret the context-free symbolic primitive elements of a program ("P") as the primitive elements within an analysis of a cognitive capacity is bound to fail because context of the conceptual elements is essential to the cognitive capacity.

I have granted that Dreyfus and Dreyfus's argument is a worthy position against classical computationalism. A system operating according to an interpretation of discrete syntactic entities as elements of a conceptual analysis of a domain as fluid as human cognition is not likely to yield a system with the same fluidity as humans.

What is of interest in the Dreyfus's position for my purposes is their expectation that connectionism is doomed to fail as a method for constructing theories about cognition. The dilemma for connectionism is well described by the Dreyfuses. If a connectionist researcher has a neural net which can successfully perform a particular task, she can attempt to develop an interpretation of the primitive elements of her model which will give her a theoretic explanation of the task (a process which we will be examining in the later section of this chapter) or she has a machine which is highly interesting physical system able to perform a task but which has no interesting theoretical

relationship to the human capacity for performing the same task. The key flaw in the computationalist understanding of cognition is the interpretive move, according to Dreyfus and Dreyfus. They expect connectionism to fail because in order to say anything about human cognition a PDP theorist must make the same interpretive move that the classical computationalist makes. But the difference between connectionism and classical computationalism is that what is interpreted is different.

I do not claim that this difference is enough to negate the argument that the cognitive domain has no theoretical structure. But I do think such a claim is too strong. Most theorists in cognitive science are quite careful to make clear that their approach to the study of intelligence may fail. David Marr (1977, 39) provides a typical example of the attitude most theorists in the field appear to take toward their approach to examining cognition: "All important fields of human endeavour start with a personal commitment based on faith rather than on results. AI is just one more example." The Dreyfuses assume that the application of a theoretical tool to a specific domain presupposes an assumption on the part of the theorist that that domain *must* have the same structure as the theoretical tool. But this assumption implies that a theorist does not pay attention to the results of trying to apply a certain theory to a particular domain. He or she blindly goes about applying a theory to the domain of inquiry because that domain *must* have the structure of his or her theory. The more likely scenario is one in which the initial "faith" (in Marr's sense) of a researcher is either dissipated or reinforced by the results of his inquires. At some point the research endeavour must be judged to be failing or succeeding. Trying to define that

point is difficult. But once a judgement of failure is made, Dreyfus and Dreyfus would conclude that the cause of the failure is an incongruence between theory and domain. But the basis for that conclusion is derived from the attempt to apply a theory. To conclude that a domain has no theoretical structure, the failure of several research endeavours, with scientific credentials, on that domain has to have occurred. Such is not the case with cognition. The scientific study of cognition is relatively new compared to the scientific study of other domains. Computationalism, as a research endeavour into human cognition, is not at the point where we can make the claim that human cognition is a domain which has no theoretical structure. We may not have the correct theoretical tools for describing the structure of the domain (if one exists), but at the philosophical level where such claims about the presence or absence of such structures are made, I part ways with Dreyfus and Dreyfus.

The model upon which connectionism bases its understanding of cognition borrows its structural features from biological accounts of the human brain. Comprised of simple processing units and their connections, a neural net resembles the biological structure of the brain: "For it has long been known that the brain is composed of many units (neurons) linked in parallel by a vast and intricate mass of junctions (synapses). " (Clark, 1989, 84). An important similarity between PDP models and the brain is their employment of simple processing units connected in a complex parallel manner.

But biological neurons do not resemble the processing units of a PDP system. PDP processing units and biological neurons are understood to have an abstract functional resemblance. What is important to remember when thinking of connectionist models is that the do not include any elements which store or use syntactic primitives in their processing units. This difference makes the processing of information in a connectionist system different form the process in a physical symbol system. The computational processes of a PDP system do not lend themselves to the manner of conceiving of cognitive capacities as being manipulations of symbols as is the case with a physical symbol system. The interpretive stage required to explain cognitive capacities is, therefore, applied to a different sort of structure than in classical computationalism.

First, let me present the defining parameters of a PDP model of cognition. Any PDP system is based on a model which has eight defining characteristic: 1) a set of simple processing units, 2) a state of activation, 3) an output function for each processing unit, 4) a pattern of connectivity among the processing units, 5) a rule for the propagation of activity-patterns through the network, 6) an activation rule for combining the inputs to a processing unit with the current state of activation of that unit which yields that unit's new level of activation, 7) a learning rule by which patterns of connectivity are modified by repeated activity within the network, and finally, 8) an environment in which the network operates.

The particular sort of connectionist models I shall be concerned with are

comprised of three different types of processing units. They are called: input units, output units, and hidden units. Each unit of each type, at any given time, has associated with it a numerical value which is called its "activation value". These numerical values are determined according to a function (activation rule) which combines a unit's present activation value with the net value of all influential connections to that unit. Each connection to a unit has a certain strength or "weight" which is also represented numerically. Weights can have either a positive or negative value depending on whether the connection is excitatory or inhibitory. The complete state of activation of a PDP system is represented by an n-place vector of real numbers; "n" being equal to the number of processing units in the system. Each layer of types of units can also be represented by a vector; each place in the vector corresponding to the activation level of one unit. So, the input layer of units can be represented by a vector with each place in the vector representing the activation level of one input unit. Input to a PDP system is begun by "... imposing numerical values on the input units of the network; these numerical values represent some encoding, or representation, of the input." (Smolensky, 1988, 1) [see also, 156-165 of Churchland 1988]. As an input vector passes through a PDP system it under goes changes in its constituent values. The net result of these changes is the output vector representing the activation values of the output units. All changes in the numerical values of the output units are a result of the functions involving activation levels and weights. Initial activation levels are determined rather arbitrarily before training begins. Training consists of repeatedly passing through the system input vectors, correcting mistakes in output vectors via some learning rule and adjusting weights throughout the

system in order to produce the correct output. After the system has settled into consistently producing the sought after output vector, the "knowledge" a PDP system employs to produce that output vector is understood to be represented in the connection weights. Thus, the patterns of connectivity within a trained neural net are supposed to represent the task-related features of the environment in which the system is located. The semantic interpretation of the PDP representational vehicles is our primary focus and I will return to this topic shortly.

But first, I want to examine the properties of neural nets which emerge as a result of the type of representation used by these systems. These emergent properties fuel the interest in continuing to research the possibility that cognition can be explained using a PDP model. One of the important features of these properties is that they resemble to an interesting extent the sort of introspectible features of our cognition (Clark 1989, 89-90).

There are four properties I want to examine. The first is called "content-addressable memory". The ability to have flexible access to internally stored information is characteristic of systems with content-addressable memory. Because any particular aspect of an activation pattern within a neural net can activate the entire activation pattern, neural networks have flexible access to stored information. In this context, flexibility is the ability to recall information related to any aspect or portion of that information.

"Graceful degradation" is the second property. The name refers to a system's ability to satisfy performance demands despite damages to the system. A system with this property is able to endure minor hardware damage without becoming completely incapacitated. Neural nets are able to operate reasonably well despite partial or erroneous data resulting from minor damage. A net will continue to operate by changing the weights between undamaged connections in the system. The capacity for making changes in the connections between undamaged processing units is a result of the third emergent property: "default assignment". Missing data is replaced by assigning particular values to operational units. The assigned values are determined probabilistically. New partial data patterns are assigned according to a "best fit" criterion in relation to stored data patterns.

The ability to complete partial data patterns of new data to match complete patterns in stored data is called the property of "flexible generalization". A net is able to generate a complete data pattern if it is given a portion of a data pattern it has stored. Having this property enables a net to generate a set of typical patterns associated with a subset of its stored data patterns; that is, if a net receives some partial data pattern as input it will generate the entire data pattern most closely associated with that partial data input.

Evidently, these properties are closely related. Each of them has to do with the ability to continue to perform a task despite non-ideal conditions. The properties emerge within a neural net as a result of its architectural characteristics, not because

of some "data-completing" algorithms or subroutines. A neural net does not have a program in this sense. The interesting aspect of these properties is that they resemble the sort of properties we conceptualize when thinking about human cognitive capacities such as memory or recognition. For instance whole memories can be triggered by the presence of only a part of their content; a song can bring to mind an entire past event in which the song played only a minor role. Also, we are able to recognise people on the basis of being able to see only their silhouetted profiles.

But if the resemblance is merely superficial between the emergent properties of neural nets and some of the introspectible properties of human cognitive capacities, it can be used as support for the claim that a neural net must match a human in complexity in order for all properties of human cognition to emerge. Dreyfus and Dreyfus (1988, 331) speculate along this line of reasoning:

> If the minimum unit of analysis is that of a whole organism geared into a whole cultural world, neural nets as well as symbolically programmed computers still have a very long way to go.
>
> Perhaps a net must share size, architecture and initial connection configuration with the human brain if it is to share our sense of appropriate generalization. If it is to learn from its own "experiences" to make associations that are human-like rather than be taught to make associations that have been specified by its trainer, a net must also share our sense of appropriateness of output, and this means it must share our needs, desires and emotions and have a human-like body

> with appropriate physical movements, abilities, and vulnerability to injury.

Turing (1950, 457) also acknowledged this line of reasoning:

> It can also be maintained that it is best to provide the machine with the best sense organs that money can buy, and then teach it to understand and speak English. This process could follow the normal teaching of a child. Things would be pointed out and named, etc.

The philosophical underpinning to the quotes above is the age-old question: what is a person? Dreyfus and Dreyfus pick out the minimum unit of analysis for computationalism as being a "whole organism in a whole cultural world". "Persons" can certainly be understood as members of this set. Turing suggests teaching a computational machine as if it were a child. If the ultimate goal of the study of intelligence is to eventually have the knowledge to be able to build a person, then the unit of analysis should be a whole organism in a whole cultural world. But the minimum unit of analysis for computationalism has never been such an organism because the goals of computationalism have never been to build a person, despite some rather "heady" predictions by some of the pioneers in the field [see Turing, 1957]. The units of analysis have been symbols in the case of classical computationalism and numbers in the case of connectionism.

In chapter one we examined the semantic interpretations applied to the symbol manipulations of classical computationalism. The interpretations are propositional and are borrowed from the taxonomy of Folk Psychology. Remember that the

representational vehicle for classical computationalism is a symbol whose semantic interpretation consists of its being a constituent element of the concepts used to inferentially analyze the cognitive task being modeled. Such concepts looked familiar since they were borrowed from the familiar family of concepts we use to make sense of each other's behaviour. These concepts are theoretically reduced to abstract functional relations between the symbols, or are somehow reside in the computational manipulations upon those same symbols. But in connectionism, it is not symbols that need to be semantically interpreted in order to explain cognitive capacities; it is numbers. The representational vehicle for connectionism is a pattern of activation numerically represented by vector transformations.

Paul Smolensky (1988) attempts to define the connectionist perspective on explanations of cognitive capacities within what he calls the "subsymbolic paradigm". Rather than analyze a cognitive task from the conceptual level, researchers using the subsymbolic paradigm analyze a task from the "subconceptual" level. Connectionism does not use the structure of natural language as a template for the theoretical descriptions of cognitive capacities. Several suggestions have been made as to what should play the role that language plays in classical computationalism. Before examining any of the alternatives suggested by theorists, I want to examine the more general issue of knowledge representation in connectionism.

2.2 Knowledge Representation in PDP systems

In chapter one we saw that computationalism necessarily implies a representational theory of mind in order to explain psychological phenomena. Knowledge is represented within a classical computational machine in a formalization which can have multiple physical realizations. The type of physical instantiation is not important. One physical instantiation may be marks on a piece of paper while another may be synaptic activity between biological neurons. The essential similarity between physical instantiations of knowledge is assumed to be located within the abstract relations that are instantiated by both. But connectionism takes a different view on the representation of knowledge.

A connectionist system is a learning system and this difference with classical machines is reflected by the different style of representation a connectionist system uses:

> A learning system should be able to acquire and update its rules automatically on the basis of its existing rules while learning to solve new problems. In practice, however, the existing rules are often unable to handle new problems, which makes the entire system very "brittle". (Hanson and Burr, 1990, 472)

Systems employing classical computational-style representations tend to be quite brittle. The introduction of novel elements to their task environment generates new problems which often stymies rigid rule-following systems. Connectionist systems do not follow rules in such a rigid sense. They are learning systems which employ

"distributed representations".

According to Hanson and Burr (1990), the fact that neural nets are learning systems are necessarily related to the fact that they employ distributed representations. As noted above, a learning system needs to be able to represent information in such a way as to be able to update and modify that information fluidly and dynamically in real time. The computational architecture able to support a learning system is a neural net with a layer of hidden units. Such an architecture has the resources available to enable the system to produce distributed representations. Hanson and Burr claim that:

> Learning is likely to produce a distributed representation of the mapping between the input and output domain, partly because of [1] the way the input features are mapped onto outputs and partly because of [2] the statistical characteristics of the data presented to the net. (1990, 477)

Condition [1] is manifest by the type of output function, activation rule, propagation rule, and learning rule chosen for a particular neural net. The second condition addresses the nature of the environment a PDP system must learn within. The production of distributed representations is dependent upon both conditions and the presence of a hidden layer of processing units between the input and output units.

Hidden units have three essential properties. Their first property is the fact that they are structurally necessary to the production of distributed representations (as alluded to above). If the successful completion of an information-processing task demands the use of distributed representations, it is the hidden units that facilitate the

satisfaction of such a constraint on that particular task. Secondly, hidden units change the "distribution" of any particular distributed representation. All learning rules for neural nets change the weights of the connections to hidden-layer processing units. Changes in weights change the distribution of a representation since changes in weights also change patterns of activation. The patterns of activation within the hidden-units are assumed to be the key to discerning what is being distributively represented in a neural net. The final essential property of hidden-units is their relation to the semantic interpretation of neural nets. The patterns of activity amongst the hidden- units determines the characteristics of the conceptual space used to perform such an interpretation of neural net capabilities.

The semantic interpretation of output and input-unit activity is relatively straightforward because they employ "local representations" of environmental features. A net is designed to produce a mapping between a very specific set of inputs and a particular output. Each input-unit transduces that, and only that, feature of the task environment it is designed to transduce. Any activity in individual input-units represents employment of a specific environmental feature, with the only variation belonging to a dimension along which a measurement of that feature's contribution to successful task-completion can be taken. The semantic interpretation of output-unit activity is equally straightforward. Output is numerically defined prior to any activity being imposed upon a net.

The mapping a net, as a whole, performs is a many-to-one relation. But each

and every processing unit also performs a mapping. Input units perform one-to-many mappings. Output units perform many-to-one mappings. An interesting feature of hidden-units is that they perform many-to-many mappings. Semantic interpretation of the patterns within the mappings performed by hidden-units is my focus here and it is also a topic about which connectionist researchers' opinions can vary.

Each processing unit can be mathematically specified as a function. The input to a processing unit is specified according to that particular unit's "fan-in" function. Let w_{ij} be the numerical value (the "weight") of the connection between unit i and unit j . The general form of a fan-in function is:

$$\sum \; w_{ij} o_j \; = \; net_j$$

(where o_j is the output value of unit j , and net_j is the value of the fan-in function used in a unit's activation function.)

A processing unit's activation function, which determines the numerical value that will represent the present activation level for a unit, has the following general form:

$$a_i(t) = F \; (a_i , net_j)$$

(where $a_i(t)$ is a real number representing the activation level of unit i at time (t)).

The output of a processing unit is specified by its "fan-out" function.

The general form of a fan-out function is:

$o_i(t) = f(a_i)$

(where $o_i(t)$ is the output value of unit i at time (t)).

The general form of the functions that each processing unit in a neural net performs indicates that a connectionist researcher needs to specify two functions within her model. At present, researchers have concentrated most of their efforts on the function, F, which combines activation levels of a unit within summations of input weights to a unit. The other function, *f*, which determines output of a unit is given several constraints. I will outline the constraints imposed upon this function before going into the various possibilities being explored for the specifications of F which go to the heart of the issue of semantic interpretation of neural net activity.

PDP models employ neurobiological research to inform their design decisions. In the case of the function employed to determine output of a processing unit, researchers look to the output of biological neurons for guidance. The output rate of presynaptic biological neurons is a function of their level of stimulation. Neurons fire, or "spike", at a rate which is a function of their stimulation levels. The rate at which a neuron fires (i.e. its spiking frequency) is most easily mathematically depicted using monotone, increasing-activation functions (Hanson and

Burr 1990, 475). But their exist other possible fan-out functions which lend themselves to more powerful statistical analyses. These analytical tools give theorists more ways for interpreting the activity in a network. At present, many possible fan-out functions can be considered but the constraint indicated above is complied with in the basis of neurological evidence. An ideal development would be a principled understanding of the relationship between various problem situations and the type of output function employed. But the research has yet to lead to this development. For now, researchers try to depict the output of a processing unit as realistically as they can.

The activation function, F, determines the transmission of activity between units in a neural net. The function is the same for each unit in a net. Individuation of nets is accomplished via the unit-specific values for the summation of input connections. The choice for a researcher specifying an activation function is between linear or non-linear functions. Non-linear functions allow for more complex network behaviour (Hanson and Burr, 1990). But a problem arises for the theorist trying to interpret the behaviour of a net employing overly complex functions:

> ..., when a particular kind of net is being designed for a particular task, its specific features are likely to be selected on ad hoc rather then principled bases, as in using a particular programming language.

> Hence, when these features are used in psychological modelling, there is always the possibility that they will have little to do with the behavioural or cognitive phenomena of interest. (473)

The interpretation of nets is vital to their relevance for explaining psychological capacities. But if the activation functions employed are too complex, or in other words, if the only way of understanding the conceptual space of the activity in a net renders a description which in no way structurally resembles the conceptual description of the task performed by the net, then the use of neural nets in explaining psychological capacities is questionable. In this sort of situation a connectionist researcher is left with a Type-2 theory for a cognitive capacity. We have already examined the problem with being able to devise only Type-2 theories as expressed by Marr and Pylyshyn. Let us now turn to the methods for semantically interpreting activity in a neural net which researchers hope to use to avoid the trap of being able to produce only Type-2 theories for cognitive capacities. The goal is to be able to view the representations located in the activation patterns of a net in such a way that the structure of that view can be discerned. But as is suggested by Smolensky, the structure of the relations amongst connectionist representations may not resemble the structure of a conceptual level view. Smolensky calls for a sub-conceptual interpretation of connectionist representations. But it is not at all clear what such an interpretation may look like, as we shall see in the next

section.

2.3 Semantic Interpretation of Neural Network Activity

We can now more closely explore the Smolenskian hypothesis that connectionist representations do not have a conceptual level description. He contends that connectionist representations mirror the sub-conceptual level employed in cognitive tasks. This sub-conceptual level can be depicted mathematically. But the relationship between the sub-conceptual level and the level at which psychological explanations can be formulated is not known. Smolensky gives three suggestions for methodologies aimed at interpreting the sub-conceptual level in such a way as to be able to construct psychological explanations from the sub-conceptual level. I will outline these suggestions below.

But first, recall that a PDP system with a layer of hidden units has local representations at the input and output layers. Between these layers, the hidden units may be supporting distributed representations. The semantic interpretation of local sub-conceptual representations is

relatively unproblematic. An output layer of units has a specific vectorial representation. The elements of an output vector represent whatever the system designers want them to represent. For example, a system designed to perform the task of initiating buy/sell orders on a stock exchange may have a two-unit output layer. The output vectors can be {0,1} or {1,0}, respectively meaning buy and sell. The designer of the net determines the interpretation of the vectors. No further interpretive analysis of units supporting local representations is required. The activity in these sorts of units is determined to be what the designers decide to make it. The more difficult task facing a connectionist theorist is the interpretation of the distributed representations within the activity of the hidden-units.

Smolensky has three suggestions for methodologies for interpreting the subconceptual level being represented in the hidden-units' activity patterns. The question he wants to answer is : "Which activity patterns actually correspond to particular concepts or elements of the problem domain?" (1988, 7). The specification of the activity within the hidden units of a neural net is achieved by using a matrix of vectors. Smolensky claims that:

> Substantive progress in subsymbolic cognitive science requires that systematic commitments be made to vectorial representations for individual

> cognitive domains. It is important to develop mathematical or empirical methodologies that can adequately constrain these commitments. (1988, 8)

A connectionist theorist must define the environment in which her system will operate. Features of this environment are represented by the input vectors of a net designed to operate within the task environment. Each element of the environment represented by the system is called a "microfeature". For a classical computationalist, the Level-1 theory of a task specifies how and what is being computed. What is being computed is determined in accordance with the conceptual analysis of a task. The features of this sort of analysis are what a system needs to represent if it is going to be able to perform the task it is designed to perform. But a connectionist does not begin with a detailed and precise analysis of a cognitive task, in terms of what a system needs to represent in order to have the capacity being modeled. "Microfeatures" are not conceptual level features specified by analysis which is carried on independent of implementation. Microfeatures are understood to develop within a network as it learns the task it is designed to perform. The creation of distributed representations, supported by the hidden units of a net, is thought to be the way in which neural nets employ certain microfeatures which a connectionist theorist must ascertain after training is complete. Determining the characterization of distributed representations in terms of microfeatures can be accomplished in three

different ways;

1) "Representational features are borrowed from existing analyses of the [problem] domain and adapted (generally in somewhat ad hoc ways) to meet the needs of connectionist modelling." (Smolensky, 1988, 8). But ad hoc adaptations of preexisting theoretical analyses of cognitive domains fall short of being considered "systematic" commitments. Also, "the needs" of connectionist modelling is here defined according to the methodology for explaining cognitive tasks as determined by classical computationalism. If connectionist theorists attempt to adapt the conceptual apparatus of classical computationalism, then connectionism will be dependent upon another research paradigm to provide the conceptual tools for explaining cognitive capacities. Such a dependence would suggest that connectionism cannot claim to provide an actual account of cognitive phenomena and would lend prima facie support to Fodor and Pylyshyn's (1988) contention that connectionism is an implementation research program for classical computationalism.

2) Another proposal for determining the characterization of distributed representations "... views the task of constraining subconceptual models as the calibration of connectionist models to the human cognitive system. The problem is to determine what vectors

should be assigned to represent various aspects of the domain so that the resulting behaviour of the connectionist model matches human behaviour." (Smolensky, 1988, 8). The trouble with viewing the problem domain of a cognitive capacity solely in terms relative to humans can be pointed out using Pylyshyn's notion of weak equivalence. If a connectionist system mimics human behaviour, we have obtained insight into only the human capacity for coping within a particular task environment. The goal of understanding intelligence will not be subserved by limiting inquiry to human cognitive capacities. But I think the trouble with such an approach can be seen more clearly if we consider the idea of calibrating a connectionist system to the human cognitive system. To be able to calibrate the level of transduction between problem domain and input to the system would require a vectorial representation of the level of transduction of the human system. Hence, we see why Smolensky calls for a systematic commitment to vectorial representations for individual cognitive domains. But should researchers seek to develop vectorial representations relative to human cognitive domains where the level of transduction may be vastly complex? Why not begin with a cognitive domain relative to a less complicated system? The lessons learned from tackling systems less complicated than humans may provide valuable lessons as researchers move towards more sophisticated models. The trouble may be a matter of beginning a research program with the most

difficult problem rather than beginning with relatively more simple problems and gradually build toward tackling the bigger problems.

3) "Hidden units support internal representations of elements of the problem domain, and networks that train their hidden units are in effect learning effective subconceptual representations of the domain. If we can analyse the representations that such networks develop, we can perhaps obtain principles of subconceptual representation for various problem domains." (Smolensky, 1988, 8). There exist a few mathematical tools for analyzing hidden unit activity. All of the tools give a theorist a way of viewing the way in which a neural net maps its inputs onto its outputs by viewing the characteristics of the mathematical space joining inputs to output. I shall concentrate on "cluster analysis" for the sake of wanting to return shortly to the more philosophical aspects of connectionism's protects for providing the theoretical foundations for a science of intelligence.

A connectionist theorist who wants to look at the way the hidden units are facilitating the mapping between input and output is faced with solving the following general problem: Given a sample of N objects or individuals, each of which is measured on each of p variables, devise a classification scheme for grouping the objects into g classes. In relation to the hidden units of neural networks: N is the number of

hidden units, p is the range of activation levels for a particular hidden unit, and g is the set of patterns of activity. More precisely, N is the number of dimensions of the mathematical space to be analyzed; p is the value of a particular point along a particular dimension, and g are the patterns present within that space determined as a result of p values. Cluster analysis partitions the mathematical space in accordance with whatever set of patterns are present within the data. It gives a theorist a picture of the divisions of the mathematical space defined by hidden unit activity. Particular divisions or patterns must then be characterized. But the characterization of these patterns up to this point is strictly in accordance with their mathematical characteristics. The same thing can be said of any of the other methods for determining the way in which hidden units map inputs onto outputs: the characterization of the patterns of activity in the hidden units, or in other words, the characterization of distributed representations residing in the hidden units, is strictly in terms of their mathematical properties.

The mathematical properties of distributed representations are not determined by a theorist in advance of designing a connectionist system. Here is a striking difference between classical computationalism and connectionism. "Connectionist methodology, ... , allows the task demands to trace themselves and thus suggest the shape of the space in a way uncontaminated by the demands of standard symbolic

formulation." (Clark 1990, 206). In the context of mathematics, the demands of standard symbolic formulation of problem spaces are that the manner of describing problem spaces is restricted to discrete mathematics. A program is an algorithm which is the discrete step-by-step, rule governed manipulation of symbols and groups of symbols. The mathematics of connectionism is not limited to discrete mathematics. The distributed representations employed by neural networks occupy mathematical spaces which are not depictable using discrete mathematics. They develop in spaces which are describable using the resources of the mathematical theory of dynamical systems; which involves continuous mathematics. So, representations over hidden units can be specified as complex mathematical entities residing in complex spaces. But knowing how they are specified is far from knowing what is being represented by that specification.

2.4 Conclusion

Smolensky suggests looking for "microfeatures" of the problem space which are mathematically tracked as they pass through a network from the input layer to the output layer. The various vector transformations performed upon an input vector would represent alterations in the microfeatures representation as it passed through the network. The problem with interpretation of neural network activity is it is not understood how to interpret vector transformations as manipulations of representations of elements of the problem domain. Invoking the notion of microfeatures only serves to point out a possibility which does not seem to have a realization. Nowhere is there a principled outline of what constitutes a micro-feature of a cognitive domain.

If a neural network is understood to be processing information, we must know what information is being represented within the network. We must be able to specify the computational states to be interpreted. The space defined by the activity of hidden units can have a computational interpretation ascribed to it. But the power of a neural network is the fact that it can model dynamical physical systems. There is an incompatablity between treating neural networks as computational systems and dynamical physical systems. Trying to have it both ways does not work. The reason for this dilemma can be illustrated by returning to Searle's challenge that semantics in not intrinsic to syntax. If a connectionist theorist ascribes a computational interpretation of

neural network activity in order to specify the aspects of the problem domain the net models, then she can proceed to provide a semantic interpretation of the computational states instantiated by the network. But by doing so, a connectionist theorist does not meet Searle's challenge for the ascription of content to the computational interpretation of neural network activity is no different than the ascription of content that classical computational theorists perform upon programs. Searle can easily challenge a connectionist to explain the way in which the information her net is processing is information *for* her neural network in the same way he does to classical computationalists who claim that their programs process information. If a connectionist theorist does not ascribe a computational interpretation upon network activity and treats nets as dynamical physical systems, then she is left without any way of relating the processes of that physical system to the notion of being an information-processing system. In the one case, the activity of a neural net is potentially semantically evaluable given a computational interpretation of that activity. But the potential is only realized by imposing a semantic interpretation from the perspective of an observer outside the system. Searle's challenge stops the relevance of this tactic for a connectionist theorist. In the other case, the activity of a neural net is not semantically evaluable. Our connectionist theorist has designed an interesting physical system that is specified according to its physical states. What would be required is an account of how

information is contained within those physical states that does not revert to first imposing a computational interpretation of those physical states. The sense of information in such an account may be sufficiently similar to the sense of information Searle invokes to discount the claims of the classical computationalists and the claims of the connectionist who ascribe computational interpretations to neural network activity. But the prospect of such a theory seems far in the future. By vaguely pointing out a possible path for connectionism to follow, I find myself in agreement with Dreyfus and Dreyfus; the computational approach to studying cognition still have a very long way to go.

Conclusion

Now much of the appeal of the original connectionist models is that they were "neuronally inspired". That is, they were supposed to model actual or possible brain, not mental processes. But if it is denied that they are models of brain processes, then the question naturally arises: What are they models of ? If they do not correspond to a mental reality, conscious or unconscious, and they do not correspond to a neuronal reality, than what evidence do we have that there is anything in actual human cognition to which the connectionist models do correspond ?

John Searle

"Models and Reality"
Behavioural and Brain Sciences 1990, 13:2, pg. 399

I want to conclude my work by bringing the connectionist approach to the study of intelligence into the perspective of Cummins' notion that property theories serve to explain cognitive capacities and Marr's distinction of these theories into Type-1 and Type-2. By doing so, the challenge for connectionism will become a little clearer.

Cummins argues that property theories, formulated via a methodology of analysis, satisfactorily explain the instantiation of particular properties by particular systems. Cognitive capacities are taken to be dispositional properties of particular systems. The analysis of a dispositional property is achieved via a functional analysis. In the case of an information-processing capacity, a functional analysis results in the capacity being specified as a function of some sort. The inputs and outputs of these functions are specified via their semantic interpretation. Cummins refers to the analysis of information-processing capacities as interpretive functional analysis. The sense of information employed in the notion of an information-processing capacity is that of the classical computationalists: symbol manipulation. The manipulation of symbols is specified via a program. Thus, programs acquire their explanatory power because they specify the capacity to perform a sophisticated information-processing task as the manifestation as a specific arrangement of less-sophisticated relatively understood capacities. The role of interpretation in the formulation of property

theories for information-processing capacities is vital. The sequence of steps from inputs to outputs must be viewed as computational; or as the manipulation of symbols which represent inputs into symbols which represent outputs. So, the classical computationalist is armed with a methodology ideal for formulating property theories of cognitive capacities. A specific capacity is analyzed as a sequence of inferences. The inferences comply with the syntactical properties of the symbols employed to represent the information being processed by a system. To have a complete computationally-based property theory of a cognitive task is to have what Marr calls a Type-1 theory. All three levels of Marr's description of classical computational must be addressed in order to relate a property theory of a computational system's instantiation of an information-processing capacity to the instantiation of the same capacity in a human.

Marr's three levels of description are applied to a system which successfully performs the task it is intended to perform:

> At one extreme, the top level, is the abstract computational theory of the device, in which the performance of the device is characterized as a mapping from one kind of information to another, the abstract properties of this mapping are defined precisely, and its appropriateness and adequacy for the task are demonstrated. In the center is the choice of representation for

> the input and output and the algorithm to transform one into the other. At the other extreme are the details of how the algorithm and representation are realized physically - the detailed computer architecture, so to speak. (1982, pp. 24-25)

Call the top level the "cognitive transition function". The middle level is the level at which this function is demonstrated to be instantiated by an algorithm or program. It is also the level at which interpretation of the computational states is performed by specifying which states serve as representations of top level components. Under the relevant semantic interpretation of these representations, the processing of what the computational states represent is guaranteed to instantiate the cognitive transition function. The bottom level describes the physical realization of the elements of level-2.

The main characteristic for a Type-1 theory is that the top level of description is complete. The theorist is able to specify what is being represented at level-2 to by returning to level-1 where it is precisely specified what is being computed and why. The representations at level-2 represent elements of the computational theory of the problem a particular algorithm solves. The content of the representations at level-2 is specified relative to the semantic interpretation of the constituent elements of the theory of computation. The theory of computation

describes a problem as the manipulation of symbols. What those symbols represent is determined by what needs to be computed which is to be specified at level-1. A complete Type-1 theory, therefore, carries the semantic interpretation of level-1 all the way through to the bottom level. The cognitive capacity thus modelled is explained as a property of a system which can be described at all three levels.

The main characteristic of a Type-2 theory is that the task modelled does not have a level-1 description. The implication of this situation is that at level-2, it is not completely understood what is being represented since a level-2 description is intended the specify the states of the system that are serving as representations and the computational processes by which these representations are manipulated. The system solves the problem posed to it, but there does not exist an account of this performance as a mapping from one kind of information to another. A Type-2 theory falls short of being a complete property theory which is the reason Pylyshyn and Marr both desire Type-1 theories. The explanatory power of Type-2 theories falls short of Type-1 theories in relation to human cognitive capacities.

Connectionist theories of cognitive capacities fall under the Type-2 category. Connectionists who have a fully trained neural net are in possession of a system which performs the task it is designed to

perform. But in order to develop an understanding of what is being computed, that is, what cognitive transition function is being computed; a connectionist theorist must first determine what is being represented by the hidden-units to facilitate the mapping from inputs to outputs. In order to consider a trained neural net as performing a cognitive transition function, a connectionist theorist must put a computational interpretation on the activity supported by hidden units. Otherwise, it is difficult to view the activity of hidden-units as solving an information-processing problem. At this point, the connectionist faces a serious choice. To ascribe the sort of interpretations to her system required to work her way up to a level-1 description of her solution to a problem, a connectionist theorist faces exactly the same sort of problem a classical computationalist faces in the form of John Searle's challenge. The information being processed is not information *for* the system, it is attributed to it by an observer. Not to interpret the activity of hidden units computationally renders connectionism as a way of modelling dynamical physical systems.

To escape the force of Searle's challenge a connectionist theorist must abandon the impulse to make their theories conform to the Marrian-type descriptions. One way to do this is to develop other high level descriptions. But the various possible higher-level descriptions of neural nets will not likely have the characteristics of explanations we

recognise and connectionism could be accused of being explanatorily bankrupt. Another way to avoid Searle is to treat neural nets as models of dynamical physical systems and hope to some day have a way of determining the information carried by a physical system. The sort of information Searle requires is very specific and rather mysterious at the same time. I do not know how it is that I process information. But I know I do it, since I am able to know what Searle means and that it means something to me without any relation to other interpretations. I agree with Searle that my ability to do so is not related to a process requiring an interpretation given to a set of representations by an outside observer. Minds may forever escape the explanatory power of a science. The fun is in the search for understanding.

Works Cited

Churchland, Paul. 1988. Matter and Consciousness, revised edition, Cambridge: MIT Press.

- 1989. A Neurocomputational Perspective, Cambridge: MIT Press.

Clark, Andy. 1989. Microcognition, Cambridge: MIT Press.

- 1990. "Connectionism, Competence, and Explanation", *The British Journal for the Philosophy of Science*, Vol. XXI, pp. 209-235.

Cummins, Robert. 1977. "Programs in the Explanation of Behaviour", *Philosophy of Science*, 44, pp. 269-287.

- 1983. The Nature of Psychological Explanation, Cambridge: MIT Press.

- and Georg Schwarz. 1988. "Radical Connectionism", Connectionism and the Philosophy of Mind. ed. T. Horgan and J. Tienson. *The Southern Journal of Philosophy*, XXVI, pp. 43-62.

Dreyfus, Hubert L. and Stuart E. Dreyfus. 1988. "Making a mind versus modeling the brain: artificial intelligence back at a branch-point", *Artificial Intelligence*, 117, no. 1, pp. 309-333.

Fodor, Jerry, 1987. Psychosemantics, Cambridge: MIT Press.

Hanson, Stephen J. and David J. Burr. 1990. "What connectionist models learn: Learning and representation in connectionist networks", *Behavioural and Brain Sciences*, 13, pp. 471-518.

Marr, David. 1977. "Artificial Intelligence - a personal view", *Artificial Intelligence*, 9, 37-47.

- 1982. Vision. New York: Freeman.

Newell, Allen and Herbert A. Simon. 1976. "Computer Science as empirical inquiry: Symbols and search", *Communications of the Association for Computing Machinery*, 19, pp. 105-132.

Table of Contents