Philosophical Perspectives, 9, AI, Connectionism, and Philosophical Psychology, 1995

CONNECTIONISM AND THE RATIONALE CONSTRAINT ON COGNITIVE EXPLANATION

Robert Cummins
University of Arizona

I. The Rationale Constraint

Cognitive Science wants to explain cognitive capacities. Capacities are dispositions, specified by articulating what Ruth Millikan (1984, p. 20) calls a law *in situ*, a law specific to a particular type of mechanism or system. A *cognitive* capacity is a disposition to satisfy some set of epistemic constraints that define "correct" or "good" performance. The capacity to play chess, to recognize faces, to find one's way home, to learn a language, and to perceive the local environment are cognitive capacities in this sense.

There is a tradition going back at least to Aristotle according to which cognitive capacities generally depend on the capacity for reason and inference. Helmholz (1856) held that perception is unconscious inference. Chomsky (1965) spearheaded the cognitive revolution with the idea that speaking and understanding a language is to be explained as the unconscious application of a theory of that language. Fodor (1975) argued that the acquisition of cognitive capacities is a species of scientific inference, the formulation and confirmation of the sort of unconscious theory whose application underlies cognitive performance on the Chomskian model.

The fundamental assumption behind this tradition is that reasoning explains cognition: Where there are epistemic constraints being satisfied, the underlying process is an inferential process. We can express this idea as a constraint on the explanation of cognitive capacities. I call it the *Rationale Constraint*. It says that you haven't explained a cognitive capacity of S—i.e., a capacity of S to satisfy epistemic constraints—unless you have shown that manifestations of the target capacity are caused in S by a process that instantiates a justifying argument—a *rationale*—for those manifestations. In the sense intended, the partial products algorithm is a rationale for the products that are computed by executing it, and a chess program, if it is any good, is a rationale for the moves it generates. Processes that generate cognitive behavior ought, in short, to be mechanized epistemology.

The argument for the Rationale Constraint is simple. A cognitive function,

as I shall understand it, is a function whose arguments and values are epistemologically related.[1] Suppose the causal process that mediates between the arguments and values of such a function is not the execution of a Rationale. Then it would seem that either (i) the capacity has been unmasked as non-cognitive, e.g., as the result of a look-up procedure (cf., Block, 1978, pp. 281-2), or (ii) we are left with no idea how the underlying causal process could guarantee that the characteristic epistemic constraints get satisfied, and explanation fails. Failure to satisfy the Rationale Constraint, in short, is either evidence that we aren't dealing with genuine cognition, or that we have an unexplained coincidence on our hands. Imagine, for example, a device that consistently generates outputs interpretable as reasonable conclusions given an input interpretable as a set of premises about some domain. There seem to be only two viable explanations. (i) The thing is reasoning. (ii) The thing is a fake: We are looking at the results of a perhaps elaborate look-up table, and not at the manifestations of a truly productive capacity. What appears to be ruled out is the possibility that a truly productive capacity to satisfy some set of epistemic constraints requires nothing like argument generation. For it seems that the selection of appropriate implications in an unbounded number of different cases would have to involve generating intermediate conclusions in an argument-like way. At least in the productive case, justified outputs appear to be a mystery (or a cosmic coincidence) in the absence of a justifying process that produces them. I'll have more to say about the argument for the Rationale Constraint as the discussion progresses, but, for now, I take it that the *prima facie* case for the Rationale Constraint is strong enough to shift the burden of justification or criticism to those would deny it.

II. Connectionism and the Rationale Constraint

Cognitive capacities are difficult to specify with any precision. No one knows how to specify a law *in situ*, satisfaction of which is sufficient for having the ability to plan a party, eat in a restaurant, or comfort a friend. Even formal domains present difficulties. What, after all, does a good chess player do? More than obey the rules, of course. But what more, exactly? It seems the only way to specify an interesting chess function is to write a chess program, i.e., to give a precise formulation of a rationale for making chess moves.[2] This "specification problem" (Cummins, 1989, p. 111ff) makes a difficulty for orthodox computationalism whose methodology is basically the same as that of any programmer: Given a specification of a function, find an implementable algorithm for computing it.[3] If you are *not* given a specification of a function to compute, you have to fall back on some form of the Turing test: make a machine that is indistinguishable from humans *in the relevant respects, when they are exercising the target capacity*. Not hopeless, perhaps, but a swamp on anyone's view. One of the reasons the study of language has become so predominant in cognitive science is that the capacities to speak, understand and acquire a natural language are (i) clearly cognitive, (ii) complex enough to be challenging, and (iii) specifiable

with a great deal of precision. A good strategy in science is to attack problems to which your methodology happens to apply.

Connectionists can, in principle, finesse the specification problem because it is possible to "train" a network to have a cognitive capacity without having even the beginning of an analysis of it; one simply needs a good training set. A successful network, however, is not, by itself, an explanation. A working network may be nearly as difficult to understand as a brain in at least this respect: No rationale will typically be discernable in the spread of activation. Orthodox computationalists, on the other hand, can only succeed by writing a program that articulates a rationale for the target capacity. Unlike connectionists, they cannot succeed as engineers yet fail as scientists.

One natural and inevitable connectionist response to this situation is to point out that the orthodox approach is hamstrung by the specification problem: Better let the network solve the problem, the argument goes, and afterwards study the working result in an attempt to figure out how the magic is done. But there is a more radical connectionist response that I want to discuss, which is to deny the Rationale Constraint itself. This is, for example, an implication of the line that Paul Smolensky and his colleagues have been developing (e.g., 1988, 1992), and it raises some issues in the philosophy of psychology that deserve to be surfaced.

The basic argument I want to consider, then, is this:

- The Rationale Constraint is incompatible with connectionism in its most interesting form.
- Connectionism is a viable framework for the explanation of cognition.
- Hence, the Rationale Constraint must be abandoned.

Of course, one person's *modus ponens* may be another's *modus tollens*: Someone (not me) may choose to see this as an argument against connectionism rather than an argument against the Rationale Constraint. However that may be, the focus of this discussion will be on the first premise, which I'll call the incompatibility thesis.

As I see the geography of this issue, there are two general lines of thought that might underlie the incompatibility thesis. I'll begin by setting them out briefly and uncritically, then turn to discussion.[4]

The semantic arguments. The fundamental idea here is that the explanation of a cognitive capacity and its specification take place on different semantic dimensions.[5] I've seen three ways of working out this idea:

Version A: When a fully distributed[6] connectionist system satisfies a cognitive function, the causal process that mediates the argument-to-value connection is defined over "sub-symbols". For present purposes, the point about sub-symbols is that they are manipulated locally by processes that have no access to the "big picture". These micro-processes operate at the single node (activation) or single connection (weight) level and therefore have no access to the distributed representations over whose semantic contents the target cognitive function is defined.

Version B: When a connectionist system satisfies a cognitive function, it computes over representations that have no interpretation in the domain in which the target cognitive capacity is specified. Whatever it is that a set of weights or an activation pattern means, neither have meanings in the semantic space in which the target cognitive function is defined. This is supposed to follow directly from (i) the claim that connectionist representation and symbolic representation are more or less incommensurable, and (ii) the claim that target cognitive capacities are specified symbolically.

Version C: A connectionist system satisfies a cognitive function only approximately. Only under idealization do the values actually computed correspond to the values of a properly specified cognitive function. The causal process that actually mediates the argument to value connection in a network therefore cannot be interpreted as a Rationale for the values of the target cognitive function because these are not the values that are actually computed.

The computational arguments. The second general line of argument that has been leveled at the Rationale Constraint is that connectionist computation has a fundamentally different form than reasoning. I've see this argument run in a variety of ways, but I think they all boil down to one of the following.

Version A: Connectionist computation is essentially a matter of discovering or exhibiting statistical correlations, whereas most rationales are not.

Version B: Connectionist systems can mimic classical rationalizers by computing over encodings of classical representations. Since the encodings do not preserve the constituent structure of the representations they encode, network computations cannot be executions of the rationales they mimic.

Let's look now at each of these lines of argument in some detail.

III. The Semantic Arguments

What I am calling the semantic arguments for the incompatibility of connectionism and the Rationale Constraint are based on the idea that connectionist representations don't represent rationales, or anyway, not the right rationales—not the rationales that rationalize the system's cognitive capacities.

III.1: The "sub-symbols" argument. The term "sub-symbols" was introduced by Smolensky (1988) to describe the semantic role of an individual unit in a distributed representational scheme. A scheme for representing a domain D is fully distributed just in case every unit involved in the representation of one member of D is also involved in the representation of every other member of D. What we have, in short, is each element in D represented by a different pattern of activation on the same pool of units. To see why Smolensky calls the individual units in a distributed scheme sub-symbols, think of each representation of an element of D as a symbol. Since the scheme is distributed, these representations will be patterns of activation over a pool of units. No single unit represents any element in D though it does make a contribution to the representation of each element in D. Thus, the individual units in the pool can be thought of as sub-

symbols, parts of a symbol, if you like.

What I am calling the sub-symbols argument is a semantic argument because it attempts to drive a wedge between the causal organization of a network and the semantic interpretation under which it satisfies a cognitive function. The crucial premise is that connectionist computation is local in a way that entails that it operates on sub-symbols but never on the symbols themselves that are the arguments and values of a cognitive function. On this assumption it would follow that connectionist systems don't execute rationales, because their computational processes are insensitive to the relevantly significant states. The assumption that the system is fully distributed amounts to the assumption that the relevant rationale cannot be defined over the contents (if any) represented by the individual units. The assumption that computation is local, however, amounts to the claim that the causal structure of the system is discernable only in terms of the interaction of individual units. The two assumptions together decouple the causal structure from the cognitively relevant representational states, with the consequence that the relevant causal structure cannot be seen as an implementation of a rationale for the cognitive function satisfied.

The sub-symbol argument evidently stands or falls with the assumption of local computation. In a nutshell, the problem with the assumption of local computation is that the causal dynamics of a network is specified as a function on activation *vectors*, not at the individual unit level. Connectionist computation consists in computing output activation vectors from input activation vectors and weights, and this is just to say that the representation computed—output vector—is a function of the input representation—input vector—and stored representations—the weights.[7] We think of the dynamics in terms of vectors because it is the entire activation vector at t, not the activation of any particular unit, that, together with the weights, determines the activation of each unit at t′, and hence the activation vector at t′. The assumption of local computation, as it figures in the sub-symbol argument, appears patently false.

And yet, the idea hangs on.[8] I think what keeps the assumption of local computation alive is the intuitive sense that activation vectors are rather artificial. Unlike velocities and accelerations, they seem to be simply a notational artifact: One speaks of activation vectors to save ink, but it is just a gimmick for talking about each individual unit in turn. In the case of velocity, for example, we use the coordinates to specify a speed and direction. Some object is actually moving at a certain speed in a certain direction, and we pick coordinates in a way that makes for convenient representation. The coordinates are conventional. In the connectionist case, however, the coordinates are not conventional, for they are the activations of individual units, and the actual values of these matters. We can, for convenience, put activation values in a standard order and treat them as coordinates, but there is no direction or net magnitude out there that we are trying to capture. We are trying to capture the individual activations.

To neutralize this intuition completely would require a healthy chunk of metaphysics that I'm in no position to provide. But we can get a sense of what

is wrong by contrasting the connectionist case with an uncontroversial case of the sort the sub-symbol argument contemplates. Imagine, then, that you are given a set of instructions for crossing a field. They specify a start position and time, and consist of a series of instructions from the following set: {go left, go right, go straight, go back, go n steps}. Every step is to be exactly one yard, and you are to take one step per second. Now if we give a lot of people instructions like this, they can be made to spell out various things on the field like a marching band at a football game. Imagine we arrange things so that they spell out proofs. These will be intelligible from the stands, but not to the individual marchers. Indeed, it is obvious that nothing anyone does is sensitive to the representations relevant to the proof; no one's actions are sensitive to the epistemic constraints whose satisfaction makes the process a proof. Causally speaking, though, the individual actions are all there are. Each person follows their instructions and that's all there is to it.

Now, we *could* specify the states of this system in vector notation: the state of the system is given by a vector whose elements are the positions of the various persons on the field. To transform one vector into another, however, we need the instructions for each marcher. Can we, as it were, aggregate these instruction sets into one function that effects the needed transformations? Of course. But—and here is the crucial point—the resulting combined function is a fraud, because the position of person A does not, by hypothesis, depend on the prior positions of any other marchers, and hence does not depend on the prior position vector. Since it is only position vectors that have interpretations in the proof, it follows that proof states don't determine other proof states, though they do predict them. Proof states are artifacts, and the "law" that predicts later ones from earlier ones is an artifact as well. (See Cummins, 1978, for the notion of an artifactual regularity.) This contrasts with connectionist systems precisely because the activation of output units does depend on the entire input vector and hence on something that *does* have a relevant representational significance.[9]

What I'm calling the sub-symbol argument, then, fails to undermine the Rationale Constraint because its attempt to demonstrate that connectionist causal structure is insensitive to distributed representation depends on the indefensible assumption of local computation.

III.2: The incommensurability argument. Version B of the semantic argument concedes that connectionist networks compute over distributed representations, but alleges that those representations are not representations of the arguments and values of cognitive functions. I'll call this argument the incommensurability argument, because it is based on the "incommensurability thesis", namely that distributed connectionist representational schemes are incommensurable with the (typically symbolic) schemes that must be used to specify cognitive functions.

As it stands, the incommensurability argument is incoherent because it assumes that connectionist systems can satisfy cognitive functions, even though they cannot represent the arguments and values of those functions. To be sure, one could reconcile the thesis that symbolic representation and connectionist

representation are incommensurable with the thesis that connectionist systems satisfy cognitive functions by abandoning the view that cognitive functions must be specified symbolically. I have some sympathy with this view. But the result leaves us with no argument against the Rationale Constraint unless we add a premise to the effect that rationales can only be specified symbolically. This premise is worth examining in some detail, for it would, together with the incommensurability thesis, make connectionism and the Rationale Constraint incompatible.

Why might one think that a rationale can only be specified symbolically? Let's begin by getting some bad reasons out of the way.

"*Epistemic constraints are defined over propositions, not over things like images.*" This just turns on an ambiguity in "proposition". As philosophers use the term, a proposition is not itself a representation but something represented, a set of possible worlds, say. In this sense, it is at least arguable that epistemic constraints are defined over propositions, for one might hold that epistemic constraints only make sense when applied to things with truth conditions.[10] Evidence, for example, is evidence for the truth of something, so a process cannot be constrained by the evidence unless that process traffics in propositions somehow. But holding that epistemic constraints are defined over propositions in this sense doesn't yield the conclusion that rationales can only be specified symbolically unless you *also* hold that only symbolic schemes can represent propositions. But surely there is no reason to believe this. A picture, for example, can hold in some possible worlds and not others just as well as a sentence.

As psychologists use the term, a proposition is a symbolic representation. In this sense of the term, it simply begs the question to suppose that epistemic constraints can only be defined over propositions. Either way you understand "proposition," then, we have no argument here for thinking that rationales have only symbolic specifications.

"*To be epistemically discriminating you have to be logically discriminating. But logical relations are defined over symbolic structure.*" Again, we have an ambiguity. If "logical relations" is understood semantically, then they are relations among propositions, and hence independent of how the propositions are represented. If "logical relations" is understood syntactically, the question is begged, since syntax is, of course, particular to a representational scheme. The relations of interest among the formulas of symbolic logic are, of course, defined over symbolic syntax. But the relations of interest among the representations in a non-symbolic system like that of Barwise and Etchemendy (1990a, 1990b) are defined over properties of those non-symbolic representations.[11]

Indeed, the existence of non-symbolic representational schemes for reasoning seems to refute outright the idea that rationales can only be specified symbolically. But the symbolist might reply that non-symbolic reasoning is limited in a way that unsuits it for cognition generally. Only symbolic schemes, they will say, allow for (i) content independent reasoning, and (ii) unbounded reasoning competencies. The alleged boundedness of connectionist competencies will come up for discussion later on when we consider computational arguments against the

rationale constraint. I'll restrict my attention here to the claim that non-symbolic representational schemes don't allow for content-independent reasoning.

There are two questions we need to ask about content independence. First, *is* reasoning content independent? And second, is it true that non-symbolic representational schemes cannot support content independent reasoning? The answer to both questions, I think, is "no."

Is reasoning content independent? In logic, we teach our students that deductive validity turns on form, not content. Although we usually temper this message with warnings about non-deductive inference and about inferences like that from being red to being colored, the central message remains that the sort of semantic relations that are central to inference can be seen to be invariant across contents. For most of us, *modus ponens* and simplification are paradigms of good reasoning, and they are content-independent.[12]

But logic, as many people have pointed out, is not a theory of reasoning, it is a theory of validity. One of the few really clear lessons of the last three decades of research in artificial intelligence, I suppose, is that reasoning needs to be domain specific to be effective, because it needs to be driven by lots of contingent knowledge—the more the better, so long as it can be efficiently accessed. While the jury is admittedly still out on this question, it surely cannot be simply assumed at the current time that human reasoning is or even can be content independent.

We needn't worry too much about it in the current context, however, because it simply isn't true that only symbolic schemes can support content independent reasoning. A now familiar example is the Galilean geometrical scheme for reasoning about relations between distance, velocity and time. In this scheme, vertical lines represent time, with time increasing from top to bottom and horizontal lines beginning at a vertical line and projecting left represent velocities, with velocity increasing from right to left. The area of the rectangle in figure one represents the distance traveled by a body that travels at uniform velocity v for a time t_1. The area of the triangle in figure one represents the distance traveled by a body that begins at rest at t_0 and achieves a velocity v at time t_1.[13] Galileo used this scheme to reason about motion, as just described. But the scheme can be used to reason about the relations of any three quantities that are related as base, height and area. And, of course, it can be used to reason geometrically, as it was before Galileo adapted it to mechanical problems.

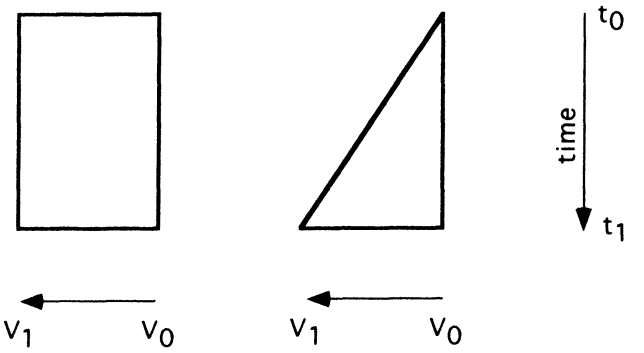

Figure one: the area of the rectangle represents the distance traveled by a body moving at uniform velocity V_1 for a time t_1; the area of the triangle represents the distance traveled by a uniformly accelerated body beginning at rest and traveling for a time t_1 when it achieves a velocity of V_1.

A final point before we leave the incommensurability argument. People do engage in symbolic reasoning. A connectionist who accepts the incommensurability thesis must somehow account for this fact. The obvious strategy here is to think of symbolic reasoning not as an explanans, but as an explanandum on a par with, or perhaps parasitic on, language use. According to this line of thought, people don't have cognitive capacities because they can reason symbolically; they rather have the cognitive capacity to reason symbolically because they can reason non-symbolically.[14] Whether connectionism can handle the capacity for language and other symbolic tasks is an open and difficult question. But it is somewhat peripheral in the current context. Given the incommensurability thesis, making connectionism *prima facie* consistent with the Rationale Constraint requires only that there be non-symbolic rationales.

The incommensurability argument, then, fails to demonstrate any incompatibility between connectionism and the Rationale Constraint. Even if we grant that connectionist representation and symbolic representation are incommensurable, there is no reason to think that non-symbolic reasoning is either impossible or an inaccurate picture of the inferential processes that underlie human cognitive capacities.

III.3: The approximation argument. Think about those XOR networks you cut your teeth on: They don't compute XOR no matter how long they are trained. First, the outputs aren't ones and zeros, but stuff like 0.9s and .11s. Here is a typical trace of performance after training:

Inputs		Values
1	1	.10
1	0	.90
0	1	.92
0	0	.09
0	.5	.76
.4	0	.59
.4	.7	.88
.3	0	.38

Table One: performance of typical XOR network after training.

What we are calling a "true" isn't even always the same number. More seriously, as the table shows, the network is quite happy with any numerical inputs, not just ones and zeros. Lines five and six of the table do not correspond to arguments and values to XOR.

"Now, then," the argument I have in mind continues, "what goes for XOR goes for connectionist computation generally. You don't get cognitive functions computed, but only approximations to them; more or less close simulations, if you like. The causal process that mediates the argument-to-value connection in the connectionist system cannot realize a rationale for the target cognitive function, for the system doesn't compute that function."

I call this the "approximation" argument because I want it to remind you of certain passages in Smolensky (1988). But the argument I've given is somewhat misnamed, for the idea is not that connectionist systems compute approximations to XOR, but that XOR can be viewed as a normalized (high numbers become ones, low numbers zeros) and restricted (only one and zero inputs count) version of the function the network actually computes. Moreover, the argument as I've presented it is self-defeating in the same way that the incommensurability argument is self-defeating. You can't argue against the Rationale Constraint on the grounds that connectionist systems cognize but don't satisfy the constraint, if part of your argument is that connectionist systems don't compute cognitive functions. We can rescue the argument from this embarrassment by supposing that, contrary to the appearances our symbolically biased epistemology generates, cognitive functions are actually like the functions connectionist systems compute in being more or less continuous.[15] Indeed, I have more than a little sympathy for this line. But it evidently lets the Rationale Constraint off the hook, unless we add the premise, lately scouted, that rationales can only be formulated for the normalized and restricted functions we *thought* were the targets. Since, as we've seen, there is no reason to believe that connectionists cannot have rationales in, as it were, their own terms, the present argument fails to show that successful connectionism is an embarrassment to the Rationale Constraint.

So much for the "semantic arguments." They don't work. The computational

arguments, though not conclusive, do better.

IV. The Computational Arguments

The fundamental idea behind this form of argument is that connectionist computation has a fundamentally different form than reasoning. Connectionist principles of computation, then, could not be principles of rationale execution. I'll discuss two versions of this line of thought, the second far more serious than the first.

IV.1: The correlation-engine argument. The thought here is that connectionist systems are, in their "learning" phase, simply correlation detectors. A trained network is then simply a correlation table with the added wrinkle of being able to automatically extrapolate between entries. No doubt connectionist networks can be used to implement other processes, but, so the argument goes, that is beside the point: At the dimension of analysis where the system is distinctively connectionist, it is simply a correlation engine. Since, however, it is obvious that rationales of the sort that rationalize cognitive performance are not typically (or ever?) algorithms for the discovery and exhibition of correlations, it follows that connectionist systems violate the Rationale Constraint.

Remember when people used to say, "All computers really do is manipulate ones and zeros,"? The reply we all learned was this: True enough, but irrelevant, because orchestrated manipulation of ones and zeros amounts to the execution of any symbolic algorithm you like. One then points out that, on the assumption (rather more suspect these days than in the golden age when we learned all this) that cognitive functions are computable functions, computers can have cognitive capacities.

It is tempting to reply to the correlation-engine argument in a similar vein: "Maybe, at bottom, connectionist systems are just correlation-engines, but who cares what happens *at bottom*." The correlation-engine argument anticipates this reply, however, by insisting that it is precisely at the dimension of analysis where the system is distinctly connectionist that it is a correlation-engine. It is conceded in advance (for the sake of argument) that connectionist systems can implement rationales of the sort that the Rationale Constraint requires. The argument insists, quite correctly, that the issue is not whether cognitive processes can be *implemented* as connectionist processes, but whether cognitive processes are properly *analyzed* as connectionist processes. The claim on offer is that rationale execution does not boil down to the discovery or exhibition of correlations. Since connectionist systems are correlations engines, either they don't compute cognitive functions, or the Rationale Constraint must be abandoned.

This argument is generally taken to be a knock on connectionism, but it could just as well be taken, by a convinced connectionist, as a refutation of the Rationale Constraint. Either way you take it, however, the argument suffers from a terminal naivete about connectionist computation. Connectionist systems do, of course, correlate inputs and outputs, but so does every computational system. The

claim that connectionist systems are correlation engines must come to more than this triviality. In the light of the fact that connectionist systems are able to solve quite general parsing problems, as we will see in the next section, there appears to be no reason to accept the bald claim that connectionist systems are *mere* correlation engines.

IV.2: The SLM argument. The correlation-engine argument is naive, but it is based on a promising strategy: Show that rationale execution and connectionist computation are fundamentally different. To establish this difference, one needs some fix on the form of rationales and on the form of connectionist computation. The idea that connectionist systems are correlation engines fills the bill nicely, because no one seriously supposes that rationales boil down to the discovery or exhibition of correlations. That idea fails because no one is in a position to argue that connectionist systems are correlation engines in the relevant sense. But there are other routes one might take to the conclusion that rationale execution and connectionist computation don't mix. The most interesting one is found in Smolensky, LeGendre and Miyata (1992; hereafter "SLM").

The cornerstone of the SLM approach is Smolensky's well-known tensor product (tp) scheme for encoding classical representations. The notion of an encoding is crucial to what follows. Let R be a representational scheme. <E, f> is an encoding of R iff f:E→R, but E does not preserve the formal structure of the elements of R. Gödel numbers are a familiar encoding scheme. Morse code, however, is not, since it preserves form, being just a different spelling of an alphabet-based scheme in which spelling is irrelevant anyway. The tp scheme is an encoding in this sense since it does not preserve the constituent structure of the classical (in the sense of Fodor and McLaughlin, 1990) representations it encodes. SLM reports development of a LISP-like programming language, TPPL, which allows for the expression of classical rationale-embodying algorithms. One takes the representations computed over by a TPPL algorithm and constructs a tp encoding. A network is then constructed, using a variation of back-propagation, that can be proved to be weakly equivalent to the TPPL algorithm. Since the tp encoding does not preserve the constituent structure of the classical representations employed by TPPL, it follows that the network does not execute a rationale defined over those representations.

SLM constitutes a serious challenge to the Rationale Constraint. It appears to establish that a connectionist system can mimic a classical rationalizer computing over representations that merely encode, but are not isomorphic to, the representations computed by that classical rationalizer. How are we to square this result with the Rationale Constraint? There are, I think, just three possibilities:

- Find some flaw in the SLM argument.
- Argue that the connectionist system does execute a rationale for the target capacity, though not one defined over the representations whose encodings the network utilizes. The network somehow finds its own rationale.

- Abandon the Rationale Constraint, and find some way to defuse the argumentation that supports it.

Is the SLM argument sound? A rationale typically defines a competence that is unbounded under idealization away from resource constraints. Idealization away from resource constraints, however, is possible only for systems whose architecture supports a distinction between the algorithm executed and the memory/time it utilizes. Consider adding machines: They do not compute plus, but a finite restriction of plus. Nevertheless, they are said to compute the full infinite function because the restriction is due to a resource limitation. Add more memory, and you relax the restriction, because the algorithm the device executes is perfectly general: It is defined for the representations of any addends whatever. A look-up table, by contrast, is inherently finite. If you are using a look-up table, adding scratch paper and patience won't help you with addends not covered explicitly in the table.[16]

The idea that a finite device can have the sort of unbounded competence typically characterized by a rationale, then, depends on a distinction between the algorithm executed on the one hand, and the resources—time and memory—available to it on the other. Schwarz (1993) argues, however, that connectionist networks do not support this distinction, and hence cannot be genuinely productive. His argument is simple. There are only two ways to add memory to a connectionist network: add units, or add precision to units already available. Since physical networks are finite, and since physical units have bounded precision, to idealize away from memory constraints in a network amounts to asking how the network would behave where more units or more precision added. If we add units to a fully distributed network that computes f, however, it will, in general, no longer compute f. For, when we add a unit, we must, in general, re-adjust all the weights. But, to alter the weights is to alter the algorithm. You haven't added memory to an existing network, the original network has been replaced by a different one. The same point holds for adding precision. To add precision requires altering the weights to allow for distinctions among previously indistinguishable activations. And altering weights amounts to substituting a new network.

A crucial and controversial assumption of Schwarz' argument is that weight change in a connectionist framework is analogous to algorithm change (reprogramming) in a classical framework, and hence that changing weights amounts to building a different system. This will seem odd to those used to thinking about networks that learn, for, when a network learns, the weights change, yet surely the network retains its identity through learning. To understand what Schwarz has in mind, therefore, it is important to distinguish learning functions, which are functions from input vectors (and, in supervised learning, targets) to weight changes, from I/O functions whose domains and ranges are sets of activation vectors. Training alters the latter functions, but preserves the former. Schwarz' assumption, then, is that we should individuate connectionist systems by I/O

function, at least for purposes of assessing the productivity issue.

The argument for this assumption appears to be as follows. Classical systems are individuated by algorithm. When we add memory to an adding machine, we don't change the way inputs are processed, we just make space for bigger ones. But when we add memory to a network, we change the way every input is processed, and this amounts to reprogramming the system. This argument is seriously flawed, however, by a misconception of classical systems. The function determined by a classical algorithm is a function from input and internal state (i.e., the state of memory), to output and internal state. It is important to see things this way because we want to be able to say that a change in stored knowledge, while it changes the input-output properties of the system, does not change the algorithm executed. When we program a classical system, we program it to behave differently as a function of different stored knowledge. This is what allows us to describe learning coherently as a change in the stored knowledge of a system that persists through that change. If we think of things this way, we shall be forced to admit that there is a sense in which, when stored knowledge is altered, the processing of every input is altered too, for, since the algorithm executed is sensitive to stored knowledge, a change in stored knowledge makes the processing of each input subject (at least in principle) to different constraints. But this is evidently not to say that changing stored knowledge amounts to writing a new algorithm and hence to building a new system.

We should think of connectionist systems analogously. If we do, we will think of a point in weight space as a point in stored knowledge space. A connectionist algorithm, then, is a recipe for computing a function from an input vector and a point in weight space to an output vector and a point in weight space.[17] From this point of view, we do not build a new network when we change weights any more than we build a new classical system when we change its stored knowledge, and this is what allows us to coherently describe learning as a change in the stored knowledge of a persisting system.[18]

The productivity issue, as we've seen, turns on whether it makes sense to idealize away from memory constraints, and, for connectionism, this depends on how connectionist networks are individuated. Schwarz is quite right in supposing that if we make identity of computational route from input to output a necessary condition of system identity, then we cannot coherently idealize away from memory constraints in connectionist systems.[19] But the argument proves too much, for, if we make identity of computational route from input to output a necessary condition of system identity, we cannot coherently describe learning in either classical or connectionist systems. Schwarz is right to point out that there is, in connectionist systems, an internal relation between how large memory is and what is in it, while the relation between memory size and memory content is external in classical systems. But you cannot, so far as I can see, promote this into an argument against productivity in connectionist systems.[20]

Schwarz' argument bears on our question only on the assumption that finite competencies—capacities that are finite even under idealization from resource

constraints—are not genuinely cognitive. The underlying idea is that finite competencies can be mimicked by a look-up table, and hence are always subject to "unmasking". I'm not at all sure we should find this persuasive. Even in the finite case, it seems that output could bear any or all of the standard epistemological relations to input and initial state, and hence qualify as cognitive behavior under the only standard of cognitiveness that we have, namely epistemic constraint satisfaction. But we are right to concentrate on unbounded competencies when discussing the bearing of SLM on the Rationality Constraint, for a network that is weakly equivalent to a *finite* TPPL rationale *might* well be simply a look-up device, hence not cognitive at all. The cognitiveness of a finite system seems to depend on the presence of a rationale. An SLM-based attack on the Rationale Constraint cannot rest on the existence of finite networks that are only weakly equivalent to TPPL rationalizers, since such an attack cannot presume the cognitiveness of a finite system while at the same time arguing that it executes no discernable rationale.

Fond as I am of the Rationale Constraint, I'm prepared to concede that the encoding argument shows that the SLM strategy works: You can design a network that is weakly equivalent to a classical rationalizer by having it compute over non-structure preserving encodings of the representations utilized by the classical rationalizer. There is, however, still a way that a defender of the Rationale Constraint can assimilate this result, for it is still open to defenders of the Rationale Constraint to suppose that systems like those proposed by SLM do in fact execute rationales, though not, of course, the same rationales as their TPPL counterparts.

To get this line of defense off the ground, it is helpful to begin with an idea of Haugeland's (1991), namely, that symbolic representational schemes and activation vector/weight matrix schemes belong to different genera of representational schemes. For present purposes, the essential point is that schemes from different *genera* are semantically more or less disjoint, only very approximate translation being possible.[21] The incommensurability argument, scouted above, made use of this idea to support the claim that connectionist systems cannot execute rationales defined over symbolic representations. We can accept this point, I argued, without prejudice to the Rationale Constraint, provided we could make room for rationales defined over activation vectors and weight matrices. But which rationales are those?

I don't know. We know so little about non-symbolic representational schemes, and our epistemological concepts are so closely wedded to the linguistically expressed and hence symbolic rationales they were developed to evaluate, that we can only speculate at present. Non-symbolic rationales are possible, as the work of Barwise and Etchemendy (1990a, 1990b) demonstrates. Taking heart from this, friends of the Rationale Constraint may take a major goal of connectionist research to be the articulation of such principles of connectionist representation as will make possible the formulation and study of connectionist rationales. In the meanwhile, those who, like me, are friendly to both the

Rationale Constraint and to connectionism, needn't be dismayed by a showing of mere weak equivalence between a symbolic rationalizer and a connectionist network, for it is demonstrable that not all reasoning is symbolic, and it is at least possible that we will, one day, be able to discern non-symbolic reasoning in the disciplined spread of activation. I hope so. It is uncontroversial that there is reasoning in the brain, and that there is spreading activation there. But it is increasingly controversial to suppose that the brain is a symbol-cruncher in (very deep) disguise. *We* are symbol users, of course, but we should recognize our symbol use for what it is: An astounding bit of cultural technology. The fact that we can use symbols no more *entails* that our brains are symbol users than the fact that we use can-openers entails that our brains use them. If you think that the brain is in the spreading activation business, and you also think that it is reasoning that explains epistemic constraint satisfaction, then you had better take seriously the possibility of a connectionist epistemology.

Why not simply abandon the Rationale Constraint? Because it is very compelling. It says simply that a process cannot preserve an epistemological virtue V (or any other, for that matter) without being sensitive in some way to V-making factors. Bugs that cannot detect drawn lines cannot consistently run drawn mazes. If you design a bug that embarrasses my principle, I will take it as a research problem, not a refutation. I will do this because it is very dangerous to abandon compelling constraints on explanation. We can always respond to a theory that doesn't deliver the kind of understanding we want by training ourselves and our students to be more easily amused, but this will only trivialize science and produce hollow scientific successes. No doubt the progress of science can and should inform our conception of scientific explanation. But we should not repeat the mistake of the deductive-nomological model of explanation and confuse predictive or deductive success with explanatory success.[22]

SLM, it is interesting to notice in passing, wears its D-N patch on its sleeve:

> Now we see that the principles of SSP [sub-symbolic paradigm] do indeed provide a non-Classical explanation for the systematicity and productivity of higher cognition. To recapitulate, less formally: the patterns of activity which are mental representations have a combinatorial (tensor product) structure which mental processes are sensitive to; the constituents in these representations figure crucially in the statement of certain high-level regularities (e.g., systematicity and productivity) in behavior; the combinatorial structure of the representations figures centrally in the explanation of this behavior (*via* mathematical deduction); but the constituents do not have causal power in the sense of figuring in mental algorithms for generating behavior: These causal algorithms can *only* be stated at a level lower than that of mental constituents, the level of individual connectionist units. (SLM, p. 44)

SLM has demonstrated (mathematically deduced) a weak equivalence, but the absence of a strong equivalence, between a classical rationale and a process of spreading activation. Should we say: *that explains how the network gets the*

correct answers? Or should we ask: *How in the name of heaven does the spreading activation get it right*? I'm inclined to think I'm not alone in thinking we should be pressing this second question, for it is this question that motivates the very considerable research devoted to understanding the so-called "hidden representations" in connectionist systems. (E.g., Hinton, 1986; Rosenberg, 1987; Sanger, 1989.)

Notes

1. Broader, or just plain different, definitions of cognitive functions are defensible. Nothing hangs on this terminological issue. I'm just interested in the class of functions, whatever they are called, that have the property indicated.
2. Part of the problem is that different players, or the same player on different occasions will do different things faced with the same board position. There is, therefore, no function from board-positions to board-positions, which constitutes *the* function chess players satisfy, in the way that there is a function from number pairs to numbers that is *the* function multipliers satisfy. But this is distinct from the problem I want to focus on here which is that there appears to be no way to formulate a chess function at all without articulating a rationale, hence no way of specifying the explanadum prior to the discovery of an explanans.
3. This is the "classic" methodology described by Marr (1982). He somewhat misleadingly calls specifying a cognitive function giving a computational theory. (It's misleading because what Marr calls a computational theory specifies a function to compute, but not how to compute it.)
4. This isn't a literature review. I've tried to cover all the arguments I know about, but I've organized the various points in my own way to facilitate exposition and discussion.
5. See Haugeland (1978) for the distinction between dimensions and levels of analysis. The fundamental point about the level-dimension distinction is that levels within a dimension are semantically homogeneous, a lower level simply a being more refined analysis of the level above, while dimensions differ in their semantic interpretations. A LISP program and its assembly code implementation are typically on different dimensions because the LISP is about lists and things listed, whereas the assembly code is about memory locations.
6. By a 'connectionist system' I shall generally mean a fully distributed system. Exceptions will be explicitly noted in the text.
7. By "input" and "output" here I don't mean just final input or output, but also intermediate input and output, i.e., input and output to on a given processing cycle.
8. I point no fingers. If the shoe fits... .
9. Some dialectics.

   (1) Later proof states do depend on earlier ones. Delay a few people, and all subsequent states will be different.

   Reply. True enough. But *strategically* delay several marchers so that they you get, e.g., "p&p" rather than "p&q" and the result will not be *relevantly* different: you won't, except by wild coincidence, get something that follows from the previous steps, you'll get garbage. This shows that the interpretation does not individuate states in a way that tracks their causal significance.

(2) In a connectionist system, the activation of unit A at t depends on *its* entire input vector. But A's input vector may be only a part of the vector that has a relevant interpretation. And,

(3) All an individual unit can know is a weighted sum of the activations of the units to which it is connected. Even neglecting differences in weights, if I am connected to two neighbors, A and B, I cannot tell the difference between both of them having an activation of 1 and A having an activation of 2 while B has an activation of 0. Hence, I am insensitive to differences in the input vector that make a representational difference.

Reply. Both (2) and (3) are quite right, but beside the point. It is the output *vector*, not some single unit, that needs to be sensitive to differences in input vectors. Moreover, there needn't be sensitivity to every difference; what's wanted is sensitivity to differences that matter. The test of whether there is enough discrimination is in performance: If performance is good enough, then so is the capacity to discriminate different representational states.

10. I am aware of at least one reason why one might deny even this. It goes back to Locke and Hume who held that epistemic relations are relations among ideas, and also (according to some recent commentators—see e.g. Owen (1993)) that ideas don't express propositions. Perhaps, then, reasoning can be understood as a process that traffics in things subpropositional, i.e., in things with satisfaction conditions rather than truth-conditions.
11. Our epistemological concepts have been developed in a symbolic framework. We want to hold people accountable to epistemological norms, and this means that those norms have to be applicable to, and articulated in, language. Moreover, decades of positivist and neo-positivist epistemology were couched explicitly in symbolic terms. Indeed, for many years, articulation of epistemological principles in the language of symbolic logic was a more-or-less explicit requirement for serious research. Non-symbolic epistemology is even rarer than non-symbolic logic. But these may be just what's required to understand human cognition at its most fundamental level.
12. That content independence inference is seen as the base case is brought out by the fact that there is a whole literature in psychology on content effects in reasoning. (See D. Cummins, under review; J. St. B. Evans, 1989.) The underlying assumption is that content effects are somehow deviations from proper reasoning. Often, they are treated outright as errors.
13. For further details, see Haugeland, 1985, pp.19-23.
14. If there is a language of thought, why do only humans have language? On the view that basic cognition is non-symbolic, this question has a straight-forward answer: language and other symbolic processing are very special achievements; they are *very hard* for brains. Hence, only the very best brains can do them.
15. This appears to be the line taken in Smolensky, LeGendre and Miyata (1992). If I understand them rightly, they argue that grammaticality is really a matter of degree, but that this is masked (to some extent, though not as much as MIT high churchers would have you believe) by the fact that the system always opts for the parsing with the maximum harmony. Like XOR networks, things only begin to look black and white when you ignore everything that doesn't reach some threshold.

16. It is important to distinguish idealizing away from resource constraints from idealizing away from attention lapses and the like. Performance may differ from competence—there may be errors, in short—because of factors like interruptions. But idealizing away from these sorts of error at most makes for correct computation of some finite function.

    Strictly speaking, to get an unbounded competence, you not only have to idealize away from resource constraints, you have to idealize away from physical breakdown as well.

17. Strictly speaking, it is a function from a point in activation space and a point in weight space to another point in activation space and another point in weight space. An input or output vector needn't specify the activation of every unit, yet the activation of any unit can affect subsequent performance.

18. It is instructive to see how the same confusion can arise in thinking about production systems. We think of programming a production system as a matter of writing productions, and this makes it seem that adding or subtracting productions amounts to building a new (though related) system. But if we think of things this way, we shall have to say that any change in stored knowledge amount to changes in the system, and it will be impossible to describe a system that learns. We should rather think of the productions we write as items in the long term memory of a system that consists of a production interpreter, a working memory, and a conflict resolution system. Construed thus, two production systems that differ only in which productions they incorporate are the same system with different stored knowledge.

19. This claim depends on the assumption that there is, in general, no point in weight space that will produce correct behavior for all real valued activations. When there is such a point, the required idealization will go through even on Schwarz's assumption about how connectionist systems should be individuated, since adding precision would not require a move away from this point. A corresponding point cannot be made about adding units, however. To see this, imagine an indefinite supply of units fully connected to existing units with zero weights. Adding units amounts to making some of these weights non-zero, and hence amounts to changes in weight space. In the special case in which old connections remain the same, and the new ones simply add scratch space, as in SIMPLIFIER (Cummins, 1991), it is arguable that we haven't built a new system, but this will hardly be the standard case.

    A possibility not considered by Schwarz is that outputs might be given as a finite but unbounded temporal sequence of activation vectors. This allows, in principle, for functions with infinite ranges. But, again, the crucial issue is whether there is some point in weight space that would, given enough time, suffice for the computation of any value in an infinite range. For many functions, such as addition, this presents no problem because the digits can be processed sequentially, with only enough memory required to handle carries. In effect, the system satisfies a finite function—sum two digits and carry—which amounts to an infinite function when the inputs and outputs are read as temporal sequences. But this trick won't work in general. Many other functions, such as speech production, will require idealization from memory constraints because early outputs in the temporal sequence can depend on later ones. (Think, for example, of verb agreement in English questions.)

20. There is another line of thought that seems to be influencing Schwarz. It goes like this. Identity of I/O function is a necessary condition of identity of algorithm computed. Changing the weights changes the I/O function computed, hence changes the algorithm, hence amounts to introducing a new system.

    We have to be careful how we think of an I/O function here. Evidently, when we add memory to an adding machine, it produces outputs it didn't produce before, from inputs it didn't accept before. If we think of I/O functions this way, then adding memory changes the I/O function of an adding machine, hence (by the argument on offer) the algorithm it executes, and this is not the conclusion Schwarz wants. To avoid this, we have to think of the I/O function as the one that would be computed but for resource limitations. But if we do that, identity of I/O function is no longer an independent condition on algorithm identity.

    As remarked above, Schwarz is apparently thinking that when we change the weights in a network, all the outputs are then computed in a new way, whereas, when we add memory to an adding machine, the computation isn't changed at all. But this is an artifact of the example. In general, when we change stored knowledge, we do change how each output is computed, in that the computational path from input to output is, at least in principles, different for each input-output pair. Imagine adding to the look-up table in an adding machine to handle 1 + 10 directly. Assuming the table is searched sequentially, and that this is the first entry, the contemplated change will alter the computational path initiated by every input. But it doesn't alter the algorithm at all.
21. Simple example: you cannot, as Locke and Berkeley agreed (*Essay concerning human understanding*, IV, vii, 9; *The principles of human knowledge, Introduction*) represent trianguality pictorially, though you can symbolically. Conversely, any verbal description of me is bound to hold in a different set of possible worlds than a picture of me.
22. We'd all love a hidden variable solution in quantum mechanics, because that would allow us to satisfy a fundamental constraint on explanation that currently goes begging. There is no hidden variable solution to be had, but we haven't all simply abandoned the constraint. It's the tension this situation generates that makes one of the main employment opportunities for philosophers in the middle of physics.

References

Barwise, J., and Etchemendy, J., (1990a) "Visual information and valid reasoning," in *Visualization in mathematics*, ed. W. Zimmerman. Washington, DC: Mathematical Association of America.

Barwise, J., and Etchemendy, J., (1990b) "Information, infons, and inference," in *Situation theory and its applications*, ed. R. Cooper, K. Mukai, and J. Perry. Stanford, CA: CSLI Publications.

Block, N. (1978) "Troubles with functionalism," in *Readings in the philosophy of psychology*, v.1, pp. 268-305.

Chomsky, N. (1965) *Aspects of the theory of syntax*. Cambridge, MA: M.I.T. Press.

Cummins, D. (under review) "Pragmatics, logical form, and human defeasible reasoning."

Cummins, R. (1978) "Explanation and subsumption." PSA 1978, 1, pp. 163-75.

Cummins, R. (1983) *The nature of psychological explanation*. Cambridge, MA: M.I.T.

Press, A Bradford Book.

Cummins, R. (1989) *Meaning and mental representation.* Cambridge, MA: M.I.T. Press, A Bradford Book.

Cummins, R. (1991) "The role of representation in connectionist models of cognition," in Rumelhart, Stich and Ramsey, *Connectionism and Philosophy*, Erlbaum, 1991, pp, 91-114.

Evans, J. St. B. (1989) *Bias in human reasoning.* London: Erlbaum.

Fodor, J. (1975) *The language of thought.* New York: Thomas Y. Crowell.

Haugeland, J. (1978) "The nature and plausibility of cognitivism," *The Behavioral and brain sciences*, 1: 215-226.

Haugeland, J. (1985) *Artificial intelligence: The very idea.* Cambridge, MA: M. I. T. Press, A Bradford Book.

Haugeland, J. (1991) "Representational genera," in Rumelhart, Stich and Ramsey, *Connectionism and Philosophy*, Erlbaum.

Helmholtz, H. von (1856) *Handbook of physiological optics*, ed. J. P. C. S. Southhall. New York: Dover Reprint, 1963)

Hinton, G. (1986) "Learning distributed representations," *Proceedings of the Seventh International Joint conference on Artificial Intelligence.* Vancouver, British Columbia.

Millikan, R. (1984) *Language, thought and other biological categories.* Cambridge, MA: M. I. T. Press, A Bradford Book.

Owen, D. (1993) "Locke on reason, probable reason and opinion." *The Locke Newsletter.* 24: 33-79.

Rosenberg, C. (1987) "Revealing the structure of NETtalk's internal representations," *Proceedings of the ninth annual meeting of the cognitive science society.* 537-554. Seattle, Washington.

Sanger, D. (1989) "Contribution analysis: A technique for assigning responsibilities to hidden units in connectionist networks. *Technical Report CU-CS-435-89.* Department of computer science, University of Colorado at Boulder.

Schwarz, G. (1993) "Connectionism, processing, memory," *Connection Science*, v. 4, 3-4: 207-226.

Shiffer, S. (1987) *Remnants of meaning.* Cambridge, MA: M. I. T. Press, A Bradford Book.

Smolensky, P., LeGendre, G., and Miyata, Y. (1992) "Principles for an integrated connectionist/symbolic theory of higher cognition," Tech. Report 92-08, Institute of Cognitive Science, University of Colorado.

Smolensky, P. (1988) "On the proper treatment of connectionism." *The behavioral and brain sciences.* 11:1-74.