MARCELLO GUARINI

A DEFENCE OF CONNECTIONISM AGAINST THE "SYNTACTIC" ARGUMENT

ABSTRACT. In "Representations without Rules, Connectionism and the Syntactic Argument", Kenneth Aizawa argues against the view that connectionist nets can be understood as processing representations without the use of representation-level rules, and he provides a positive characterization of how to interpret connectionist nets as following representation-level rules. He takes Terry Horgan and John Tienson to be the targets of his critique. The present paper marshals functional and methodological considerations, gleaned from the practice of cognitive modelling, to argue against Aizawa's characterization of how connectionist nets may be understood as making use of representation-level rules.

Kenneth Aizawa has argued against the view that connectionism can be construed as an approach to cognitive modelling which makes use of representations which are processed without the use of representation-level rules (Aizawa 1994). The representation without rules (RWR) approach to cognitive modelling has been defended by Terry Horgan and John Tienson in a number of works (Horgan and Tienson 1988, 1989, 1991, 1992, 1996, 1999a, b). In fact, even Aizawa has gone on to argue that *some* systems may well qualify as having representations without rules (Aizawa 1999, 74–85). However, unlike Horgan and Tienson, he makes it clear that he does not think any *connectionist* approach to cognitive modelling can legitimately claim to be an RWR approach. He bases the preceding claim on what he calls the "syntactic" argument. The purpose of this paper is to rebut Aizawa's argument.

While I will defend the possibility of construing connectionism as an RWR approach to cognition, my characterization of RWR and my arguments are quite different from the Tienson and Horgan approach to defending RWR. Part one of this paper contains (a) a presentation of Aizawa's syntactic argument, and (b) an explanation of the relationship between my position and that of Tienson and Horgan. Part two contains a functional response to Aizawa's syntactic argument, and part three contains a methodological response.

Synthese **128:** 287–317, 2001.

1. SOME BACKGROUND AND AIZAWA'S "SYNTACTIC" ARGUMENT

As I understand it, the RWR approach is committed to the following claims:

(1) cognitive systems make use of representational structures subject to semantic interpretation, and

(2) there are no representation-level rules governing the processing of semantically interpretable representations.

Aizawa's interpretation of RWR (which focusses on the Horgan and Tienson construal of the position) is less general than the preceding (Aizawa 1994, pp. 466–468). For example, (1) does not require that representational structures be either semantically or syntactically structured (as Aizawa requires). It allows for the possibility that they are syntactically structured but not semantically structured,[1] or for the possibility that they are neither semantically nor syntactically structured.

Claim (2) is *not* intended as a denial that there are rules at work in or that are useful for describing a system. In other words, the idea is *not* that it is improper to describe the processing of individual neurons or synapses in a network as being governed by mathematical rules. Nor is it improper to say that there are representation-instantiation (RI) rules which specify how to interpret the representations being processed. The point is that systems (connectionist or otherwise) are not best understood as following rules having representational content of the sort expressed by natural language sentences. While the (1) and (2) are different from Aizawa's construal of RWR, they do not beg any questions against his syntactic argument. I am sure he would agree that his syntactic argument should apply to (1) and (2), at least insofar as (1) and (2) are alleged to be realized in connectionist nets. I say "alleged" because the whole point of the syntactic argument is that RWR cannot be realized in any connectionist net.

It is worth pointing out that (1) and (2) constitute a more general construal of RWR than what Horgan and Tienson defend. My purpose here is to lay out the broadest possible construal of an RWR approach and to show why the syntactic argument does not work against it. However, before preceding further, a few words should be said about the relationship of my position with that of Horgan and Tienson, who have argued that human cognition does not *conform* to programmable, representation-level (PRL) rules. Rather, they claim, cognition is best characterized by the use of soft laws (or laws with ineliminable same-level exceptions). They go on to assert that connectionist networks are capable of realizing the defeasible causal tendencies captured by soft laws, and that these networks are not best understood as conforming to PRL rules (Horgan and Tienson 1996, 95–96; 1999a, 12–14). Aizawa has argued that the preceding is not so. However, he has not only argued that connectionist nets *conform* to or are *describable* by PRL rules, he claims that his syntactic argument shows that connectionist nets *execute* such rules. In fact, he argues that because connectionist nets can be understood to execute representation-level rules of the sort he specifies, it follows that they are rule describable (Aizawa 1994, 468).[2] The planets may be said to conform to the laws of motion, but they are not generally said to execute or make use of them to compute their orbits. After presenting Aizawa's syntactic argument, I will argue that there is some sense in which connectionist nets cannot be said to execute representation-level rules, which supports the assertion that there is a legitimate sense in which connectionist nets can be said to process representations without representation-level rules. This defence of the RWR position is quite different from the defence found in the work of Horgan and Tienson, who want to defend the much stronger claim that human cognition (and the neural networks realizing it) do not even *conform* to representation-level rules. The evaluation of that position is beyond the scope of this paper. I direct the reader to a recent exchange between Aizawa and Horgan and Tienson for further discussion on that topic (Aizawa 1999, Horgan and Tienson 1999b). What follows in the rest of this paper is (a) a presentation of Aizawa's argument that connectionist nets do execute representation-level rules, and (b) a critique of that position. Consequently, while I am defending a certain type of RWR construal of connectionism, I cannot be understood as defending the much stronger RWR construal of connectionism offered by Horgan and Tienson.

Aizawa asks us to consider a deterministic, feed forward neural network having only four nodes; two nodes are input units, and two are for output. The network computes these input–output mappings with the following interpretations (where "→" is read as "maps to"):

00 → 00
01 → 11
10 → 11
11 → 11

and let the RI_i and RI_o rules be the following,

$RI_i(00)$ = no dog is present
$RI_i(01)$ = a dog is present on the right
$RI_i(10)$ = a dog is present on the left
$RI_i(11)$ = a dog is present on the right and the left

$RI_o(00)$ = do not start fleeing
$RI_o(01)$ = start fleeing
$RI_o(10)$ = start fleeing
$RI_o(11)$ = start fleeing

Then, the rule forms are instantiated as follow

> If no dog is present, do not start fleeing,
> If a dog is present on the right, start fleeing,
> If a dog is present on the left, start fleeing,
> If dogs are present on the right and the left, start fleeing.
> (Aizawa 1994, 476–477).

The subscripts indicate whether the representation instantiation (RI) rule applies to an input or an output.

Three conditions are laid down for the interpretation of such four node networks (Aizawa 1994, 476). First, if an input pattern has no semantic interpretation, then there is no representation-level rule describing how it is processed. Second, if two formally distinct input patterns, such as $RI_i(01)$ and $RI_i(10)$, receive the same semantic interpretation, then the representation-level rules are modified. There are two cases here: (a) if the formally distinct but semantically equivalent inputs are mapped to the same (semantic) output, then we have identical representation-level rules, and one be eliminated from our list; (b) if the formally distinct but semantically equivalent inputs are mapped to semantically different outputs, then the two representation-level rules generated using the above strategy are replaced with a single rule having the same antecedent and a disjunction of the two consequents. The third condition is that the representation-level rules are understood as applying in parallel. "Of course, these rules might be implemented serially, but that is mere implementation" (Aizawa 1994, 476), or so he thinks. The first two interpretive conditions do not apply to the network just described, but an example is given of how to apply them (Aizawa 1994, 477–478).

The conclusion from all this is that for any feed forward, two-state, deterministic four node network having the configuration of two input nodes connected to two output nodes, it is possible to come out with a list of representation-level rules describing the processing in those cases where the inputs and outputs are semantically interpretable. This sort of argument is then repeated for four node probabilistic networks (Aizawa 1994, 479–481). Aizawa suggests that it is obvious how to extend the argument random two-state and k-state networks, and goes on to extend the argument to networks with hidden units. He points out that either the hidden units have a semantic interpretation or they do not. If they do have a semantic interpretation, then the above argument can applied from the input layer to the hidden layer to derive a set of representation-level rules, and it can be applied from the hidden layer to the output layer to derive another set of representation-level rules. If there are many hidden layers with semantically interpretable activation patterns, then the above strategy is simply iterated from one semantically interpreted layer of nodes to the next to yield a set of representation-level rules to describe the processing. If hidden units do not have a semantic interpretation “then they may be relegated to the status of implementation detail and ignored” (Aizawa 1994, 481). Apparently, what all of this is supposed to show is that if connectionist nets have semantically interpretable inputs and outputs, then they can be understood as having representation-level rules governing their processing (Aizawa 1994, 479, 482–483). In more recent work, Aizawa has gone on to sketch out what he thinks could legitimately be called an RWR approach to cognition, but he reaffirms his position that connectionist nets cannot be thought of as RWR systems (Aizawa 1999, 75).

The first observation I would like to make about this argument is that it is oddly named. So far as I can see, it does not depend on syntax. What is required is that there are semantically interpretable inputs and outputs, but that need not require that they be syntactically structured. Indeed, it is a trivial matter to imagine a two node net, with a single input node and a single output node, for which we could run Aizawa’s argument for the generation of representation-level rules, and there is not much room for syntactic structure when using single nodes for inputs and outputs. However, rather than fuss with what this argument should be called, I prefer to turn to its evaluation.

My objection is simply this: it is one thing for a system to be rule describable, but quite another for it to actually make use of causally efficacious rules. The orbits of the planets are rule describable, but the planets do not make use of or consult rules in determining how they will move. In other words, planetary motion may *conform* to rules even if no rules are *executed* by the planets. At best, the syntactic argument shows that connectionist nets conform to or are describable by representation-level rules; it does not show that they execute representation-level rules. To his credit, Aizawa anticipates this objection and points out that those who are inclined to make it must provide an account of what it is to make use of a rule which allows for a system to be characterized as having RWR[3] (Aizawa 1994, 479). In the following sections, I plan to sketch out strategies for responding to this challenge. It is important to note that Aizawa does not take himself to have proven that such task is impossible. I take his argument as an attempt to shift the burden of proof on to those who want to argue that connectionist nets can be properly characterized as having RWR. My account of what it is to make use of a rule (in the required sense) will not be complete, but what I hope to do is shift the burden of proof back on to those who would argue that connectionist nets cannot be characterized as having RWR.

2. THE FUNCTIONAL RESPONSE

In this portion of the paper, I will lay out Martin Davies' conception of tacit knowledge of a rule, modify it, and apply it to connectionist nets to suggest that there may very well be a sense of what it is for a system to make use of a rule which allows many connectionist nets to be understood as having RWR. I will defend this use of tacit rule following by showing that it explains a number of intuitions we have about the functional properties of systems. If this should prove insufficient to the reader, part three of this paper will marshal methodological considerations in favour of the sense of rule following defended in this section; the methodological considerations will be shown to be closely related to the functional considerations.

Classical cognitive modelling is committed to being able to differentiate between behaviourally equivalent systems which are following different sets of rules (Davies 1989, 541–547). One of Quine's criticisms of Chomskian linguistic theory is that there will always be more than one set of axioms from which we can derive a body of linguistic behaviour. Given the preceding, what sense can be given to the claim a linguistic agent is making use of one set of axioms as opposed to another empirically equivalent set? Clearly, cognitive modellers owe us a story as to what it is to be following one set of rules as opposed to another set which leads to the same behaviour. Davies has argued that the notion of having knowledge of a tacit rule needs to be developed to respond to the Quinean concern. He suggests that the way to do this is to have the derivational structure of a theory reflected in the causal processes of a system. Much of the rest of this section is an attempt to develop the preceding idea.

Let us begin a thought experiment about a drinks machine (Davies 1991). It is a simple device accepting four kinds of input tokens. Each input token produces a different kind of output. The following table summarizes the input to output relations.

Input Token	*Output*
red square	coffee with milk
blue square	coffee without milk
red round	tea with milk
blue round	tea without milk

Clearly, either kind of square token will produce coffee; either kind of round token produces tea; red tokens produce milk, and blue tokens produce a "black" beverage. Davies asks us to consider two ways in which the insides of this machine may be layed out. In the first layout, there are four, totally autonomous mechanisms, one for processing each kind of token and producing the respective beverage. In the second layout, there are three mechanisms which work together – one which produces coffee if it senses a square token, another which produces tea if it senses a round token, and a third which adds milk to the beverage produced if a red token is sensed. Davies uses these different layouts to explain what he calls causal systematicity of process. Roughly, a process is causally systematic relative to some semantically characterized input-output pattern G if and only if there is some common causal process mediating the transition from the input state to the output state. If we ask whether the drinks machine is causally systematic relative to "Given some sort of request for coffee, provide coffee", our answer will depend on the layout of the machine. Layout one is not causally systematic with respect to such a pattern, but layout two is. In layout one, there is no single causal process involved in the production of all and only coffee outputs. In layout two, there is a mechanism that deals with all and only requests for coffee and provides the appropriate outputs for those requests. Davies uses the notion of causal systematicity of process to give an account of what it is to have tacit knowledge of a rule. Roughly, a system has tacit knowledge of a rule G* only if a physical state mediates all and only state transitions conforming to some pattern G in virtue of one and only one type of formal feature tokened by all inputs falling under G, where G is either a semantic-semantic characterization or a semantic-action characterization of an input-output pattern (Davies 1991, 236–238). The drinks machine can be used to clarify that claim. If we want to know whether the drinks machine has tacit knowledge of the rule "Provide tea given a request for tea", we must ask if there is some formal property shared by all requests for tea which engages a mechanism that processes all and only requests for tea. Layout one does not possess such a mechanism; layout two does. Consequently, layout one cannot have tacit knowledge of the rule in question, whereas layout two may. By "tacit knowledge", Davies has in mind the kind of knowledge which an entity can be said to have even if he, she, or it is not conscious of having the knowledge or cannot make it an object of second order reasoning. For a system to have tacit knowledge of a rule is to say that it can follow that rule even if it does not know that it follows that rule. Moreover, this tacit knowledge may be either explicitly represented or hardwired into the system. However, the fact that a rule is explicitly represented or encoded in a system does not mean that the system can represent that rule to itself, reason about it, or be conscious of it.

Davies claims that connectionist nets tend not to be characterized by causal systematicity of process, which is to say that they are not characterized by the possession of tacit knowledge of rules. I am inclined to agree. However, Davies' account of tacit knowledge of a rule is in need of some clarification and elaboration. Let us start by re-examining the account stated above.

> D1: A system has tacit knowledge of a rule G* only if a physical state causes all and only state transitions conforming to some pattern G in virtue of formal feature f[4] tokened by all inputs falling under G, where G is either a semantic-semantic or a semantic-action characterization of an input-output pattern. G is the extension of (or the set of ordered pairs picked out by) G*. (Henceforth, the account of G will be omitted when G is referred to in discussions about tacit rules.)

D1 will not do, since a system may have two rules of the same type being processed at the same time (or in parallel). In other words, a single physical state in a system may not mediate *all* tokens of state transitions characterized by G. For example, a computer may solve a problem by breaking it down into subproblems, solving the subproblems simultaneously (in different parts of the machine), and bringing the solutions to the subproblems together to construct a solution to the original problem. In solving a problem in this way, different parts or states of the machine may be carrying out the same type of instruction. This means that one physical state token will not carry out *all* tokens of a particular type of processing in the type of system under discussion. This problem might lead to the following, modified necessary condition for a system having tacit knowledge of a rule.

D2: A system has tacit knowledge of a rule G* only if at least one physical state token P
(a) causes only state transitions conforming to some pattern G, and
(b) has the potential[5] to cause a token of every type of state transition conforming to G in virtue of formal feature f, tokened by all inputs falling under G.

Notice that (b) does not claim that every token of every type of state transition conforming to G will be caused by a particular formal feature, and that is what allows the same rule type or state transition type to be realized in different parts of the system either at the same time or at different times.

Unfortunately, D2 will not do as an account of what it is to have tacit knowledge of a rule. To see why, imagine that we wrote two programmes for von Neumann machines. Each takes textual representations as input and provides text as output. The inputs, outputs, and the mapping of inputs to outputs for the programmes will be the same as the inputs, outputs, and mappings of the drinks machines. Machine language is used to implement the programs, which we will say are written in BASIC. Some of the BASIC instructions for the first program include the following.

```
If Input$ = “square and red” then print “coffee with milk”.
If Input$ = “square and blue” then print “coffee without milk”.
If input$ = “round and red” then print “tea with milk”.
If input$ = “round and blue” then print “tea without milk”.
```

Some of the instructions for the second program include the following.

```
If input1$ = “square” then print “coffee”.
If input1$ = “round” then print “tea”.
If input2$ = “red” then print “with milk”.
If input2$ = “blue” then print “without milk”.
```

The first programme is meant to be analogous to the first layout of the drinks machine mentioned above, while the second programme is meant to be analogous to the second layout. Call the computer running the first programme *drinks computer one* (DC1) and the computer running the second programme *drinks computer two* (DC2). In these computers, none of the above rules will be implemented in a single state. Instead, each of the rules will be translated into low-level machine language rules. Each of those rules will be sensitive to the binary translations of the textual inputs. DC2 may be said to have tacit knowledge of the rule

R1: Given a request for coffee, provide coffee

even though there is no one state which has the potential to cause all transformations in accordance with that rule in virtue of only one type of formal feature. This means that D2 needs to be revised.

D3: A system has tacit knowledge of a specific rule only if it has either *simple tacit knowledge of a specific rule* or *compound tacit knowledge of a specific rule*.

A system has simple tacit knowledge of a specific rule G* only if it has at least one physical processing state token P which
(a) causes only state transitions conforming to some pattern type G, and
(b) has the potential to cause – in virtue of causal sensitivity to formal feature f, tokened by all inputs to P falling under G – a token of every type of state transition conforming to G.

Q is a set of physical state tokens P1, P2, ... Pn. A system has *compound tacit knowledge* of a specific rule G* only if it has at least one set of physical processing states Q such that
(i) the states in Q cause only state transitions conforming to some pattern type G;
(ii) the states in Q have the potential to cause – in virtue of causal sensitivity to formal feature f, tokened by all inputs to Q falling under G – tokens of every type of state transition conforming to G, and
(iii) when input conforming to a pattern G is provided to Q, all states in Q will carry out processing.

This is better, but it requires some explanation.

Consider drinks computer one (DC1). Such a system could not be said to have simple tacit knowledge of the rule

R1: Given a request for coffee, provide coffee.

The reason is that there is no state or set of states which has the potential to cause all state transitions in conformity with the rule and to do so in virtue of causal sensitivity to only one type of formal feature. The reason is that the inputs "square and red" (coffee with milk) and "square and blue" (coffee without milk) both constitute requests for coffee, but they have different sets of states processing them. DC1 cannot even be said to have compound tacit knowledge of R1 since there is no single set of states which has the potential to cause all state transitions in accordance with R1. In other words, there is no single set of states which, when they are all brought to bear on either type of request for coffee, will produce the correct coffee output. Condition (iii) plays an important role in the preceding argument. Without that condition we would have to say that DC1 has compound tacit knowledge of R1. The reason is that DC1 can be understood as satisfying conditions (i) and (ii) of D3. There is a set of processing states which appropriately processes "square and red" inputs, and there is a different set of processing states which appropriately processes "square and blue" inputs. If we take the union of the preceding sets of states as forming the set Q, then conditions (i) and (ii) are satisfied with respect to the rule R1. However, DC1 does not have tacit knowledge of R1 since there is no common causal process which operates on all the inputs falling under R1 – hence the need for condition (iii). To say that a state or set of states expresses tacit knowledge of a rule is to say that there is some common causal process which operates on all inputs falling under that rule. The exercise of moving from D1 to (D3) was an attempt to refine that intuition, and condition (iii) is essential to capturing commonality of causal processes. Without it, DC1 would qualify as having compound tacit knowledge of R1 even though it has one set of causal processes for operating on some inputs falling under R1, and a different set of causal processes for operating on other inputs falling under R1, and no one set of causal processes which has the potential to operate on all inputs falling under R1. While DC1 does not have any kind of tacit knowledge of R1, it does have compound tacit knowledge of

> R2: Given a request for coffee with milk, provide coffee with milk.

The reason is that all the conditions specified by D3 are satisfied. There is a single set of processes which operate on all inputs falling under R2.

According to D3, DC2 has compound tacit knowledge of R1 but not R2. DC2 has compound tacit knowledge of R1 in largely the same sort of way that DC1 has compound tacit knowledge of R2. As we saw in the subset of programme rules above for DC2, there is a specific BASIC rule which corresponds to R1. Therefore, there is one set of processing states which has the potential to process all and only inputs falling under R1. The same thing can be said about

R3: Given a request for milk, provide milk

and

R4: Given a request for no milk, do not provide milk.

DC2 uses R1 and R3 to process requests for coffee with milk, and R1 and R4 to process requests for coffee without milk, but there is no set of states used to process all and only requests for coffee with milk. This is why DC2 fails to meet the conditions specified by D3 for having tacit knowledge of R2.

One last thing should be noted about the notion of a tacit rule (as I have used it and as Martin Davies has used it). It straddles the distinction between explicit and hardwired rules. I plan to use the notion of a tacit rule to draw a distinction between classical and connectionist processing. While many would concede that a connectionist net does not have explicit rules, they might question why it cannot be said to have hardwired rules in much the same way that a classical machine might. Consequently, it is not enough to show that there are no explicitly represented rules in a connectionist net to show it is non-classical.

Thus far, I have only attempted to suggest a plausible necessary condition for having simple or compound tacit knowledge of rules. There may very well be other types of rules of which we have tacit knowledge, and this is something to which I will return below. At this point, something needs to be said about why only a necessary condition has been stated. The truth is, stating both necessary and sufficient conditions is no easy matter. I refer the reader to an exchange between Martin Davies and John Searle on the issue of tacit rules (Davies 1995a; Searle 1995; Davies 1995b).[6] For the purposes of this paper, though, I can get by with a plausible necessary condition. What I want to show is that connectionist and classical processing have important functional differences which are connected to important methodological differences. If a plausible necessary condition can be stated for the kind of rules followed by classical systems, and if it can be shown that connectionist systems do not satisfy that necessary condition, then a plausible difference will have been identified between these types of systems. It is now time to put forward this argument in more detail.

Imagine a deterministic network with four input units and two output units which carried out the following mappings:

Interpreted Input	*Input Coding*		*Interpreted Output*	*Output Coding*
square and red	1010	→	coffee with milk	11
square and blue	1001	→	coffee without milk	10
round and red	0110	→	tea with milk	00
round and blue	0101	→	tea without milk	01

Aizawa is committed to saying the this network is executing the following rules in parallel:

T1: if given a square and red input, produce a coffee with milk output.

T2: if given a square and blue input, produce a coffee without milk output.

T3: if given a round and red input, produce a tea with milk output.

T4: if given a round and blue input, produce a tea without milk output.

The problem with this strategy is that other sets of rules can be said to be followed by this network. Consider the following:

T5: if given a square input, produce a coffee output.

T6: if given a round input, produce a tea output.

T7: if given a blue input, produce a milk output.

Let us use "S1" to designate the set T1 through T4, and "S2" to designate the set T5 through T7. The problem with identifying a set of rules which is compatible with the performance of a network and then saying that the network executes those rules in parallel is that it is too easy. The network described above could be understood as executing all members of S1 in parallel or all members of S2 in parallel.[7] Aizawa's approach picks out S1, but why should that be privileged over S2? One possible response is that it does not matter. The network can be understood as executing S1 in parallel or S2; as long as the behaviour predicted by both sets is equivalent, there is no substantive issue. This is a logically possible response, but it will not do if the practice of cognitive science is to be taken seriously. Cognitive scientists *do not* say that any grammar which predicts a body of linguistic behaviour is an adequate account of the grammatical rules which are being executed in the head. Presumably, grammars are supposed to supervene on brain states; change the rules of the grammar, and there should be a change in the brain states encoding the grammar. In other words, if the Quinean critique of Chomskian linguistics is to be averted, there darn well better be a way of differentiating between different grammars (or sets of rules) which lead to the same linguistic behaviour. This commitment of classical approaches to cognitive linguistics is largely implicit, but a commitment it is. If two grammars lead to the same input-output mappings, then to say they are different grammars is to say there are functional differences of some sort between the grammars. Similarly, to be able to distinguish a system which uses S1 from a system which uses S2, there must be functional differences (or differences in how the rules interact with each other, inputs, and outputs) between the systems. Of course, for most everyday or folk uses of intentional ascription, it makes no difference whether we ascribe to an agent beliefs corresponding to the contents of S1 or beliefs corresponding to the contents of S2. But for the purposes of ascribing tacit rules in cognitive modelling, we must differentiate between S1 and S2.[8] The inability of Aizawa's approach to do this in a principled and well motivated manner suggests that it is inadequate.

The network described above does not have tacit knowledge of specific rules. Perhaps it has some sort of tacit knowledge, but it is not best characterized as being *of* specific rules. Roughly, there is no state causing all and only state transitions picked out by any one of the T_i; more precisely, none of the T_i satisfy D3. Connectionist modellers have come up with a number of techniques for understanding how a network does what it does, so it is not a mystery that networks can do the kinds of processing that they can. That processing is simply not best understood in terms of executing representation-level rules. The whole point of the RWR approach to cognitive modelling is that it is possible to process representations without the use of rules. The network processing drink requests has representations for inputs and outputs, but it does not appear to have representation-level rules processing those requests.

The argument of the preceding two paragraphs rules out the possibility that the drinks net is processing the members of S1 in parallel or the members of S2 in parallel. However, it should be made clear that none of the above precludes the possibility of some sort of system which executes tacit rules in parallel. A system which executes the members of S2 in parallel will contain a set of physical states ST5 which carries out T5, a different set of states, ST6, which carries out T6, and still another set of states, ST7, which carries out T7. For the ST_i to be executed in parallel, the following conditions must be satisfied:

(a) $ST5 \neq ST6$; $ST5 \neq ST7$; $ST6 \neq ST7$;
(b) each of the ST_i satisfy D3, and
(c) the rules T5, T6, and T7 are executed simultaneously.

In other words, the same physical state(s) cannot be said to be processing T5, T6 and T7, for the reasons outlined above. Of course, this does not rule out different sets of physical states sensitive to different inputs and causing different state transitions. In other words, exactly the same set of states cannot all be said to participate in processing T5, T6, and T7, but different sets of states corresponding to each rule may realize these rules and engage in processing simultaneously.[9]

One objection which may be raised at this point is that some classical systems seem to follow rules which are not captured by the account of tacit rules given by D3. For example, some chess playing computers may try to get the opponent's queen out early even though nothing like "get the opponent's queen out early" is represented in the system. Since such behaviour emerges from other rules and is not encoded according to D3, it might be objected that D3 is inadequate. This would be a mistake. D3 is intended to help us distinguish between different sets of rules which may lead to the same input-output behaviour. That D3 does not sanction "get the queen out early" as a tacit rule is a virtue. If we expanded D3 so that it included any rule which correctly describes a system, then we would have a conception of rule execution which would be incapable of responding to the Quinean objection to Chomskian linguistics, and that is not the conception of rule execution cognitive modellers are interested in. The point of D3 is to capture a necessary condition of the conception of rule execution at work in classical cognitive modelling so that it may be argued that such rules are not at work in connectionist nets.

Another objection which may be raised at this point is that the notion of tacit rule developed thus far is quite restricted. It has nothing to say about rules which do not make reference to a particular type of input. Thus far, only tacit rules which are sensitive to specific formal features have been discussed. Is it not the case that the following might be realized as a tacit rule?

R5: "Given any regular verb, add 'ed' to form the past tense".

Yes, I think an account must be given of what it is for a system to realize R5 and rules like it as a tacit rule. D3 states a necessary condition for a system having simple or compound tacit knowledge of *specific rules.* More work needs to be done on stating conditions for having tacit knowledge of general rules. However, the purpose of this paper is not to provide a complete account of the different ways in which a system may be said to have tacit knowledge of a rule. My primary aim is to suggest that some connectionist nets fail to have tacit knowledge of specific rules, rendering plausible the idea that a system may process representations without the use of representation-level rules. It might be objected to the preceding that while connectionist nets may fail to have tacit knowledge of specific rules, they might turn out to have tacit knowledge of general rules, rendering some sort of RWR interpretation of connectionism implausible. Something needs to be said to assuage this concern.

Consider the drinks net. What if someone were to say that it does have tacit knowledge of a general rule.

R6: For every understandable coffee request, provide the appropriate coffee output.

Why not say that the drinks net has tacit knowledge of R6 since the synapses and output neurons cause all and only state transitions conforming to R6? If we can say this, then the drinks net is not an RWR system since a representation-level rule is mediating the processing. My main concern with this sort of objection is that it does not adequately capture what is at issue in the debate over the plausibility over RWR systems. Not once does Aizawa resort to such *general* rules to argue that connectionism makes use of representation-level rules. Such a strategy is very simple – too simple. As I understand PRL rules, they are either specific rules (characterised by D3) or are realized using *specific* rules. The preceding is an important point, for if (on a complete analysis of tacit knowledge of a rule) the drinks net can be said to have tacit knowledge of R6, it does not have this tacit knowledge in virtue of implementing specific tacit rules, which is very different from classical cognitive modelling, where general representation-level rules are carried out using specific representation-level rules. This is why I take the present work to a rebuttal of Aizawa which may be used for a qualified defence of the RWR approach; it is not a knock-down defence of the idea that connectionist nets are properly characterized as not having representation-level rules of any type. However, I do not think this trivializes the project. Consider this: when people wonder whether NETtalk is making use of representation-level rules, they are not interested in whether it is following a single rule such as, "Given any English text as input, provide the relevant speech as output". They are interested in whether there are many specific, representation-level rules at work, such as those at work in DECtalk, the classical predecessor of NETtalk. Also, not having tacit knowledge of specific rules has important methodological implications, as we will now see.

3. THE METHODOLOGICAL RESPONSE

If cognitive science was only interested in which inputs to the central nervous system get mapped to which behavioural outputs, then we would not have to worry about how the inputs get processed. However, since such a crude behaviourism has been shown to be profoundly limited in what it can explain, cognitive science has become interested in explaining how inputs are processed. Classical models of processing tend to make use of processing states having simple or compound tacit knowledge of rules to explain how inputs get converted to outputs. It turns out that the rejection of tacit rules not only has explanatory consequences, but methodological consequences as well. Classical methodology tends to take a top-down[10] approach to cognitive modelling. The notion of a tacit rule can be understood as playing an important role in that methodology.

Marr, Chomsky, Newell and Simon, and many others conceive of explanation in cognitive science as being many-levelled. According to the official story, explanation can take place at three levels. The first or highest level specifies the function to be computed; the second-level specifies the algorithm which will be used to compute the function, and the third level explains how the algorithm is to be implemented in the hardware. Many algorithms can compute a single function, and there are many ways of implementing a single algorithm. Christopher Peacocke (1986) has identified a lacuna in the official characterization of this three level story, a lacuna which will help us to clarify the classical strategy for cognitive explanation. There is a practice in cognitive science of doing research at a level which specifies more than the function to be computed but does not concern itself with the specific algorithm which an entity or system is using. Peacocke uses the example of arriving at the depth D and the physical size P of an object from the retinal size R. There are different ways to do this. An entity could simply store all the possible admissible values of P, D and R in a table. An alternative strategy would be to compute the required values by drawing on the information that $P = D \times R$. There are many algorithms which could be used to retrieve information from a look-up table, and there are many algorithms which could be used to realize a system which follows the rule (or utilizes the information that) $P = D \times R$. So it is possible to specify the function to be computed and something about how it is to be computed without being committed to a specific algorithm. Such a level of description and explanation provides more than the official characterization of level one explanation but less than level two. Peacocke calls it level 1.5. This is the level of what traditionally has been called *competence theory.* Chomsky uses competence theory in linguistic research to specify a "framework of principles and elements common to attainable human languages" (Chomsky 1986, 3). He also says that it helps to "guide the search for mechanisms" (Chomsky 1986, 221). As Clark points out, the task of competence theory is twofold. First, it is part of the appropriate explanatory and methodological strategy of cognitive science. A high level account of the phenomena in question is sought before algorithms are actually written to realize this more abstract account. Such an approach requires us to know in advance of actually writing the algorithms what sort of constraints (besides computing the appropriate function) these algorithms must satisfy. Second, as a result of the first feature, competence theory is suggestive of the kind of algorithms which should be written. Hayes' work in Naive Physics is a good example of the methodological and explanatory primacy of the competence level. Naive Physics is the research program which attempts to formalize the knowledge required by mobile beings to get around in their environment and manipulate it. Hayes' work on the naive physics of liquids included fifteen states for liquids and 74 axioms, expressed in the predicate calculus, for the movement, change, and geometry of liquids. He is quite explicit that we ought to seek a competence theory before trying to write algorithms which enable a system to move around in and manipulate its environment (Hayes 1984, 3). Given that different systems or entities may make use of different algorithms for getting around in their environment, a competence theory allows us to see what these different systems and their algorithms have in common. By working at a level of abstraction and generalization which is above the algorithmic level, competence theory allows for the specification of a revealing equivalence class of algorithms. The importance of the revealing nature of the equivalence class requires further elaboration.

As Chomsky remarked, and as is evident in the practice of classical cognitive science, competence theory is *suggestive* of the algorithms which will be written. There is a sort of close connection between the competence level and the lower levels of explanation, and the revealing nature of the equivalence class should be explicated in terms of the closeness of the connection between the competence level and the lower levels of explanation. (This will become clearer as we go along.) As we saw above, one could use an algorithm to examine a lookup table to arrive at values for P given values for D and R. One also could use different algorithms for computing P by making use of the equation that $P = D \times R$. If a competence theory specifies that a system is making use of the rule or principle that $P = D \times R$, then lookup algorithms are ruled out as possible realizations of the competence theory. It is not enough for algorithms to compute the same function for them to belong to the equivalence class picked out by a given competence theory; otherwise, lookup algorithms would belong to the equivalence class. Following Andy Clark, I suggest that we make use of the notion of a tacit rule to decide which algorithms are picked out by competence theory as members of an equivalence class. While Clark uses Davies' account of a tacit rule, we will use the more refined account expressed by D3. "A standard competence theory posits a set of rules or principles of derivation defined to apply to a class of structured, symbolic representations according to their form" (Clark 1990, 203). We can now state in some detail the conditions under which algorithms belong to the equivalence class picked out by a competence theory.

E1: Algorithms belong to an equivalence class of algorithms picked out by a competence theory if and only if (a) they compute the same function; (b) the representations over which the rules or principles of the competence theory are defined are explicitly represented in the realization of the algorithm, and (c) the rules or principles are either explicitly represented, hardwired, or both in accordance with D3.

As we saw above, the role of competence theory is two fold. First, it provides a high-level account of what different algorithms, computing the same function, have in common. It is also a key part of classical methodology; the search for a good competence theory often precedes the task of writing programs to realize the competence in question. Second, a competence theory suggests ways of realizing the rules and representations it refers to. Lookup tables have never been considered good realizations of a given competence, except in those cases where the competence in question just is the ability to search a list. Conditions (b) and (c) from E1 (or something close to them) have traditionally underwritten what a good realization of a competence theory would consist in. D3 is key in both of the roles played by competence theory. It is part of what allows us to see different algorithms as belonging to an equivalence class, and it is part of what suggests how the competence theory is to be realized. It is important to notice that connectionist modelling rejects classical competence theory and the notion of tacit rules. This appears to be doubly problematic for the connectionist. First, there would appear to be no high level explanatory account of the sort that a classical competence theory provides. Second, there is no level of theory which suggests ways of realizing a competence under study. In short, by abandoning classical competence theory, connectionism appears to abandon any attempt to satisfy the two functions served by competence theory.

I suggest that a shift in perspective is in order if we are to properly understand connectionist methodology. Clark calls this shift a sort of "Copernican revolution in our picture of explanation in Cognitive Science" (Clark 1990, 213). Consider NETtalk, a connectionist net for taking representations of text as input and producing an output which, when fed into a speech synthesizer, produces speech sounds. The network has a set of hidden units and begins training with random values on the weights. By back-propagating error, the network gradually learns to pronounce English text. To be sure, some background knowledge is brought to bare in laying out the net. Decisions must be made about how to code the input and what the phonemic output should be, not to mention how the output should be coded. However, this is nowhere near a competence theory; no rules or principles are specified for how the inputs are to be converted to outputs. If we want to understand how the network does its processing, we must look at its synaptic weights. Different sets of weights may yield the same input to output mapping. This appears analogous to the classical claim that different algorithms may yield the same input to output mapping. Indeed, the ability of a competence theory to specify an equivalence class of algorithms may even have an analogue in the NETtalk project. The technique of cluster analysis may be used to examine the hidden units. Cluster analysis is simply a technique for statistically examining hidden unit activation vectors. Each hidden unit vector is paired with its closest neighbour. The pairs are then averaged out, and each averaged pair is paired with its closest neighbour. The process can be continued until there is only one vector. The result of carrying out this analysis can be seen in Figure 1. Cluster analysis reveals how the first set of weights has partitioned the hidden unit activation vector state space. Notice how the system has learned the distinction between consonants and vowels. Exactly how a system will partition its hidden unit activation vector state space(s) to solve a problem is not generally known before actually training a net. An important fact about networks is that different sets of synaptic weights may have the same cluster analysis; this is analogous to the claim that different algorithms may be expressions of the same competence theory. Just as there is a many-one mapping from algorithms to a classical competence theory, there may be a many-one mapping from sets of weights to a cluster analysis. Before we get carried away with these similarities between classicism and connectionism, we should notice that there is an important difference between classical competence theory and connectionist attempts to explain a competence: while classicists *start* with competence theory and descend to the algorithmic and implementational levels, connectionists start by training up a net *and then* analyse the net to figure out how it is doing what it is

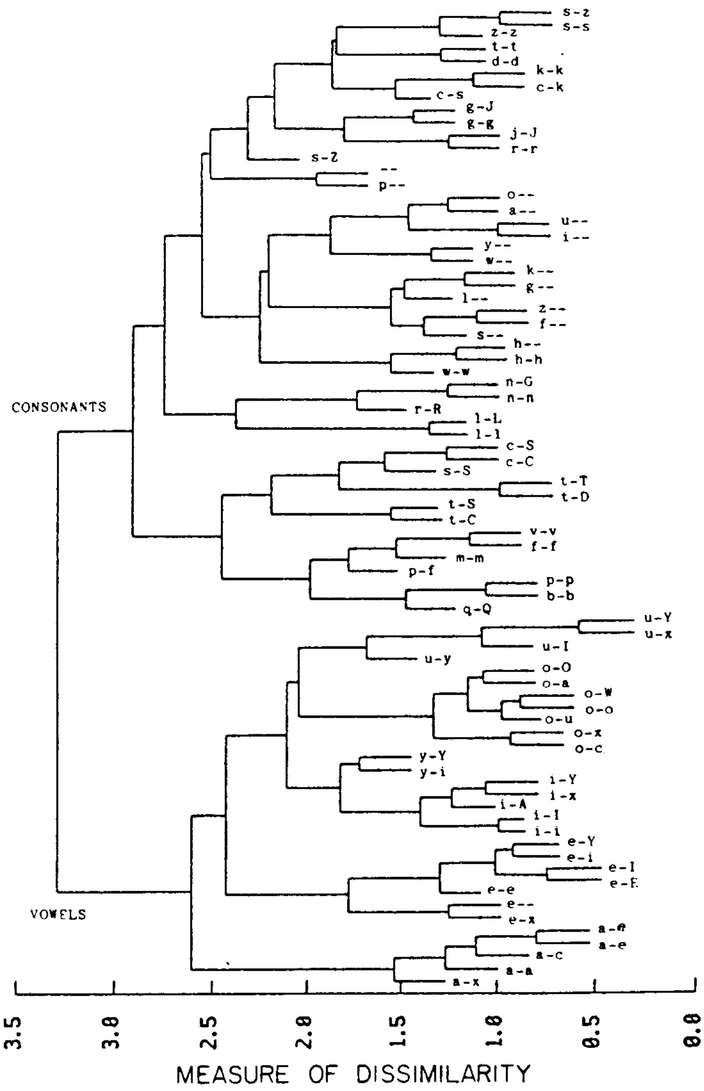

Figure 1. The hierarchy of partitions of NETtalk's hidden unit vector space shows that the network has learned to differentiate between vowels and consonants, and between specific vowels and specific consonants. It is possible for different sets of synaptic weights (or different networks) to partition the hidden unit vector space in the same way. (This figure is reprinted from Sejnowski and Rosenberg 1986.)

doing (or to figure out why it is not doing what the designer wants it to do). This is a significant methodological inversion, the "Copernican revolution" Clark writes about. For connectionists, if a competence theory is possible, it is something to be discovered by examining trained networks. Connectionists need not abandon the kind of high level general understanding given by a classical competence theory, but that kind of understanding does not play the same role in connectionist research. Above, we noted that classical competence theory is suggestive of the types of algorithms to be written and the way they will be implemented, thanks largely to conditions (b) and (c) of E1. There is none of this in connectionist attempts to explain the competence of a net. This is not surprising since the search for a higher level understanding comes *after* the net has achieved what it was designed to achieve.

It should be pointed out that cluster analysis is not the only technique available to connectionists for studying how a network does what it does. A discussion of some other techniques is already in the literature (Clark 1993, 52–67; Smolensky 1995). We need not go into them here, but it should be admitted that cluster analysis is not always a useful tool. However, even when it is not useful – for example, in certain kinds of simple, recurrent networks – other techniques may prove very useful, such as principal components analysis.

Some qualifications are in order. Clark seems to think that by rejecting tacit rules, one is committed to a sort of bottom-up methodology. This is false. David Marr's work on vision makes this very clear. He formulates certain high-level principles which are realized in the structure or geometry of a network of processors. Those high-level principles are not realized as tacit rules (at least, not according to my account), yet Marr's methodology is undeniably top-down. As an example, let us consider Marr's solution to the false target problem. Given two retinal projections of an image, how do we figure out which points on one image correspond to points on the other image? See Figure 2. The dark circles are correct matches, and the empty circles are false targets. Out of the sixteen possible matches, how do we determine which four are correct? Marr formulates the following three rules for solving the false target problem (for black and white images):

> Rule 1: *Compatibility.* Black dots can only match black dots.
> Rule 2: *Uniqueness.* Almost always, a black dot from one image can match no more than one black dot from the other image.
> Rule 3: *Continuity.* The disparity of the matches varies smoothly almost everywhere over the image. (Marr 1982, 115)

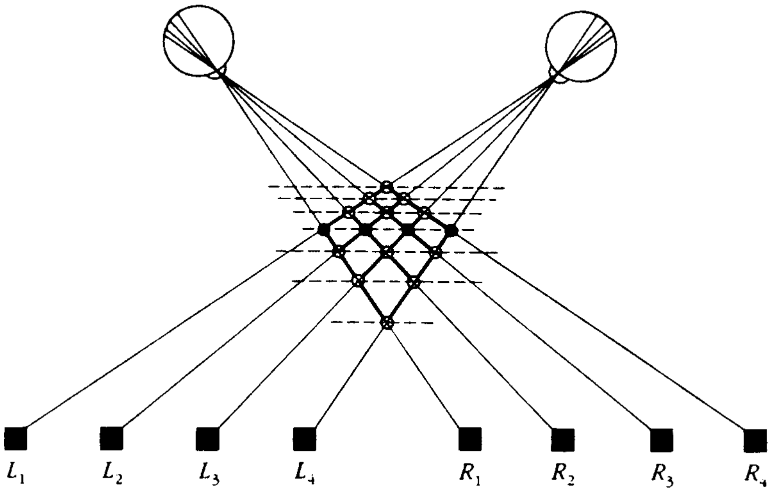

Figure 2. The false target problem.

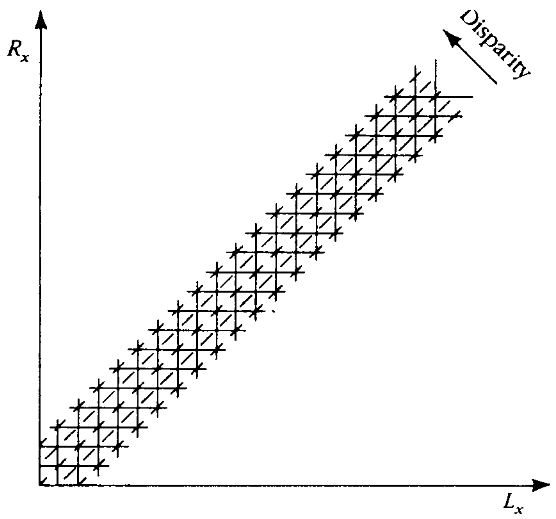

Figure 3. A network for stereo disparity. (Figures 2 and 3 are reprinted with permission from T. Poggio, ‘Cooperative Computation of Stereo Disparity’, *Science* **194**, 283–287. Copyright 1976, American Association for the Advancement of Science.)

He goes on to show how these rules can be realized in a network of processors (see Figure 3). The vertical lines are lines of sight from the right eye, and the horizontal lines are lines of sight from the left eye. The dotted diagonals are lines of constant disparity. Essentially, the second rule tells us that only one match is allowed along each vertical line and each horizontal line. The third rule tells us that correct matches lie along the dotted diagonals. Say that we insert a processor at each intersection in

Figure 3. If the processor ends up taking a value of one, then it represents a match; if it ends up taking a value of zero, then there is no match. We realize rule two by making all the vertically and horizontally connected processors inhibit one another; rule three is realized by making the diagonal connections between processors excitatory. We start the network by giving each processor a value of one if it is a place where black dots could match (which, at this point, includes false targets). Each processor adds up the ones in its excitatory neighbourhood, adds up the ones in its inhibitory neighbourhood, multiplies each sum by a weighting factor, and subtracts the inhibitory value from the excitatory value. If this result exceeds a certain threshold, the processor takes on a value of one; if it does not exceed the threshold, the processor is set to zero. This process is repeated until the processors produce an output value of one only if they represent a correct match. This network does not realize rules one through three as tacit rules since, roughly, there is no state or set of states which does all and only the processing appropriate to rule one or to rule two or to rule three. What all of this means (among other things) is that the rejection of tacit rules in cognitive modelling is not sufficient for rejecting a top-down methodology, but Clark sometimes writes as if it is (Clark 1990, 211–212). Modellers who reject tacit rules, like connectionists, tend to make use of a bottom up methodology; however, Marr has shown us that it is possible not to use representation-level tacit rules in modelling and still make use of a top-down methodology. This is why his work has both a classical and connectionist flavour to it. Methodologically, it is top-down, which tends to be the classical approach. Ontologically, he tends to be more connectionist, which is to say that his models do not use representation-level tacit rules. However, *and this is a very important point*, while the rejection of tacit rules does not always commit the cognitive modeller to a bottom-up methodology, we can see why there is a close connection between rejecting the use of representation-level rules satisfying D3 and the rejection of top-down methodology. It is one thing to take a top-down approach and keep the processing in agreement with, say, three rules; it is quite a different thing to keep the processing in agreement with 100 or 1000 rules while not realizing the rules by making use of tacit rules. Imagine trying to come up with an intuitive diagram (akin to Figure 3) that would let us see how it is possible to carry out processing in accordance with 1000 or more rules. That would be extremely difficult, to put it mildly. The greater the number of rules that processing must conform to, the more tempting it is to realize them as tacit rules when one is taking a top-down approach to modelling. Similarly, imagine trying to straightforwardly select (without some sort of trial and error method) synaptic weights in a connectionist

net that would be required to keep the processing in agreement with 100 or 1000 representation-level rules. Once again, the prospects are not good.

Earlier in this paper, I claimed that I would offer two arguments in support of the claim that connectionist nets tend not to have tacit knowledge of rules, which I would take as support for the view that connectionist nets tend not to be best understood as making use of representation-level rules even if representational content is ascribed to inputs and outputs. I referred to the first argument as the functional argument. The material we have just examined allows us to present a second argument. Even if one insisted on claiming that connectionist nets contained representation-level rules in spite of the functional considerations marshaled in part two of this paper, one would have to concede that the sort of rules connectionist nets have are not, in some sense, *programmable* rules.[11] Classical methodology in rule-based AI and rule-based cognitive modeling often starts with a competence theory which suggests how to program a system with a set of rules which will allow that system to exhibit a particular competence. What the considerations of the last several paragraphs suggest is that by rejecting the implementation of representation-level rules in accordance with D3, connectionists are often forced into rejecting the top-down, program approach to cognitive modeling. In general, it just is not clear how to realize many different representation-level rules by superposing them on the same processing states. Consequently, connectionists are generally forced into using learning algorithms that discover the appropriate synaptic weights.

If we understand a programmable rule as the sort of rule which can be realized in such a way as to make programming in accordance with a top-down classical methodology possible – which suggests that specific tacit rules are *usually*[12] realized in accordance with D3 and general tacit rules are usually realized using specific rules satisfying D3 – then we can say that connectionist researchers do not generally make use of programmable rules. While I do not claim to have a complete account of what programmable rules are, I think some non-trivial things have been said about how to characterize programmable rules, and that characterization invokes the role they play in the methodology of classical modeling. The rejection of programmable, representation-level rules provides further support for the view that connectionist nets tend to be usefully characterized as making use of representations without rules.

Some clarifications are in order. I have not claimed that a programmable rule and a tacit rule are the same thing. One might argue that Marr's three rules for solving the false target problem are, in some sense, programmable rules. At least, nothing I have said rules that out. His rules fit into the classical top-down methodology, and there is a straightforward way of realizing them while working from within that methodology. However, Marr's realization of the three rules violates D3, which means that the system he laid out does not have tacit knowledge of these rules. I take my argument in part two of this paper to be sufficient for establishing that there exists some important sense in which some neural nets can be usefully characterized as having representations without rules, where the rules in question are what I call tacit rules. It may very well be that the correct thing to say is that some neural nets, such as Marr's net for solving the false target problem, do not have tacit knowledge of rules even if, in some sense, they make use of programmable rules. However, as I have already suggested, violating D3 makes it very difficult to adhere to top-down methodology. In those cases where the violation of D3 seems to force the adoption of a more bottom-up approach, we have a reason for saying that the system in question does not have programmable, representation-level rules, which provides an additional reason (over and above the argument mounted in part two) for saying that some neural nets are best characterized as having representations without rules. In those cases where the violation of D3 does not lead to the abandonment of the top-down methodology, the functional arguments marshaled earlier in this paper provide reason for saying that such nets are not following tacit rules, and that is significant since a number classical research programs are committed to such rules.

4. CONCLUSION

A large part of what is at issue in the RWR debate is how best to understand the similarities and differences between certain kinds of classical and connectionist processing. Horgan and Tienson argue that PRL rules are a commitment of classicism, and that by rejecting such rules, connectionist nets may be better able to model cognition than classical systems (Horgan and Tienson 1999a, 9–12). They go so far as to say that the neural nets embodying cognition cannot even be understood as *conforming* either to PRL rules or to psychological laws without ceteris paribus clauses. While I am not prepared to go that far, I am prepared to suggest that many connectionist nets are not best understood as *executing* representation-level rules. Aizawa told a story according to which connectionist nets in general could be understood as being governed by or as executing representation-level rules, and I challenged the story by making use of functional and methodological assumptions at work in the practice of cognitive modeling. My challenge suggests that the issue is about more than whether it is logically possible to describe a system as following representation-level rules; it is about whether it is productive (or helpful in making sense of the practice of cognitive modeling) to do so. It might turn out that not all the details of my story are correct, and I look forward to hearing from those who would press me on the details. However, while details are important, the main goal of this paper is to shift the discussion away from whether there is some conceivable sense in which a neural net could be said to follow a representation-level rule. The issue, I propose, is whether it is *useful* in understanding the science of cognitive modeling to propose that connectionist nets follow rules. I have tried to argue that the notions of a *specific tacit rule* and a *programmable rule* help us to understand the commitments of classical cognitive modeling, and many connectionist nets cannot be understood as executing those types of rules or having those commitments. One way to respond to this is to argue that I have not properly construed the preceding notions and that a proper construal will defeat my conclusions and reinstate Aizawa's. Another way to respond to my argument is to say that there exists some other kind of representation-level rule which connectionist nets can be said to follow. Either way, the response should make reference to the practice of cognitive modeling, since what is really at issue is the suspicion that the practice of connectionists is importantly different from that of classicists, and that "the something different" has something to do with rules. Broadly construed, this paper has been an attempt to clarify the preceding point.

ACKNOWLEDGEMENTS

I thank William Demopoulos, Ausonio Marras, Wayne Myrvold, and William Rapaport for their comments on the early stages of research which lead to this paper. I am also greatly indebted to the anonymous referee from *Synthese* whose comments on my original submission helped me to improve the final version of this paper.

NOTES

[1] It might seem odd that I want to leave open the possibility that representations in the head may turn out to be syntactically but not semantically structured, but Steven Schiffer provides some interesting reasons for why one might subscribe to such a view (Schiffer 1991). I am not endorsing this view; rather, I am suggesting that because such a view has an interesting defence, we should construe the RWR approach as broadly as possible so as to not exclude from it views which clearly belong to it. While Schiffer does not defend an RWR approach, someone might want to adopt the view that the representations processed by connectionist nets are syntactically but not semantically structured, and that

such representations are processed without representation-level rules. Aizawa's characterization precludes the preceding from falling within the RWR approach, but I see no reason why it should, hence my more general construal of the approach.

[2] It is important to note that Aizawa's use of "rule governed" is equivalent to what I mean by "rule execution". In other words, he contrasts *rule describable* with *rule governed*, whereas I contrast *rule describable* (or *rule conforming*) with *rule execution.* There is no difference in substance in our distinction, only in terminology. To understand the structure of his argument (outlined in Aizawa 1994, 468), one must be clear on how he uses the expression "rule governed".

[3] Actually, Aizawa words this point differently. I have taken the liberty of rephrasing his point to free it from the Tienson and Horgan vocabulary and render it more general.

[4] What the formal feature is will depend on the system in question. It may be a pattern of activation across neurons, a pattern of activation across transistors, the colour of a physical token, the shape of a physical token, and so on. In most cases, it is obvious what the formal feature is. It is conceivable, though, that disagreement may arise as to exactly how the formal feature should be characterized in a particular system. Such disagreements would have to be dealt with on a case-by-case basis.

[5] At least as far back as Aristotle philosophers have been discussing the nature of *potentiality.* I do not claim to have (a) a detailed analysis of what it is or (b) a general way of stating how we know when potentials will be actualized. However, I do not take that to be a serious problem for my purposes. In designing a particular type of machine, computer scientists know what the potentials of certain states are, and they know the conditions under which the various state potentials may be actualized. If they did not know such things, computer design would be impossible. So while I have not given an analysis of potentiality, I do not think that I am referring to anything which is terribly controversial in the practice of computer design or cognitive modelling.

[6] For the purposes of this paper, I remain neutral as to the type of content (broad or narrow, intrinsic or extrinsic, and so on) which may be ascribed to tacit rules. A full account of tacit rules needs to say something about content, but the complexities involved in tackling this issue put it beyond the scope of this paper.

[7] This problem generalizes to feed forward networks with more than two layers. The reason is that all the processing states (synapses and hidden units) in such nets work on all the inputs to those states, so there can be no processing states which apply only to specific inputs (yielding tacit knowledge of specific rules). It is true that different inputs may lead to different patterns of activation across hidden layers, but these activation patterns are output and input states, not processing states. D3 defines tacit knowledge of specific rules in terms of processing states.

[8] This might seem to some to be an unfair move, but it is perfectly legitimate provided that one separates the purposes of everyday folk psychology from the purposes of cognitive modelling. This is not to say that there is no relationship between the two, but it is perfectly plausible to suggest that everyday folk psychological ascriptions of attitudes operate at a higher level of generality, at a level where many of the details that cognitive scientists concern themselves with tend not to be at issue. Folk psychology is not concerned with the details of how representations are processed because such details are not required for our everyday predictive and normative purposes. However, cognitive modelling does concern itself with such descriptive details, and I am suggesting that debate over RWR is not about folk psychological ascription but about a level of description which is concerned about the details of how representations are processed.

[9] Another way of understanding this point is to say that tacit rules cannot be said to run in parallel if they are superposed in the sense defined in Tim van Gelder (1991, 43).

[10] The discussion of top-down versus bottom-up methodologies oversimplifies things, but it is useful in helping to capture differences between many classical and connectionist research programmes. The point is not that connectionists do not bring any high-level background knowledge to the process of training a network; the point is that they do not bring the kind of detailed knowledge characteristic of a competence theory (a set of detailed rules to be realized) to the training of particular networks.

[11] Horgan and Tienson argue against the notion of PRL rules by using the notion of *intractability*. This is *not* what I am doing. Rather, I am pointing out that the notion of a programmable rule is something that fits into a top-down methodology, and that such a methodology is often unavailable to connectionists because of the type of systems they use. This argument depends on intuitions about what sorts of rules can be realized in a "straightforward" or "principled" manner into neural nets of various types. Compilers and interpreters are an important part of classical methodology; write a well-formed program in a well-formed language, and it is guaranteed to be realized provided memory limitations are not exceeded. Connectionist learning algorithms are much different; maybe they will find a set of synaptic weights to perform a particular task, and maybe they will not. And if they do not, then you have to modify the network architecture, or the training algorithm, or the way the inputs are coded, or some combination of the preceding, and then you try again. There is nothing very principled or straightforward about any of that. I regret that I do not yet have a highly formalized account of what the straightforward nature of realizing program rules consists in, but I believe the examples used in this paper suggest that it is a useful notion.

[12] Since it is *usually* (and not *always*) the case that abandoning tacit rules often leads to abandoning top-down methodology, it has to be admitted that E1, stated earlier in the paper, may not exactly capture the conditions under which different algorithms fall under an equivalence class. If the equivalence class E1 was meant to capture is taken to be of programmable rules, and if such rules deviate from tacit rules, then E1 is best understood as a statement of what *usually* falls within the equivalence class.

REFERENCES

Aizawa, K.: 1994, 'Representations without Rules, Connectionism and the Syntactic Argument', *Synthese* **101**, 465–492.

Aizawa, K.: 0000, 'Connectionist Rules: A Rejoinder to Horgan and Tienson's Connectionism and the Philosophy of Psychology', *Acta Analytica* **22**, 59–85.

Chomsky, N.: 1986, *Knowledge of Language: Its Nature, Origin and Use*, Praeger Publishers, CN.

Clark, A.:1990, 'Connectionism, Competence, and Explanation', *British Journal for the Philosophy of Science* **41**, 195–222.

Clark, A.: 1993, *Associative Engines: Connectionism, Concepts, and Representational Change*, MIT Press, A Bradford Book, Cambridge, MA.

Davies, M.: 1989, 'Connectionism, Modularity, and Tacit Knowledge', *British Journal for the Philosophy of Science* **40**, 541–555.

Davies, M.: 1991, 'Concepts, Connectionism, and the Language of Thought', in W. Ramsey, S. Stich, and D. Rumelhart (eds), *Philosophy and Connectionist Theory*, Lawrence Earlbaum, Hillsdale, NJ, pp. 229–258.

Davies, M.: 1995a, 'Tacit Knowledge and Subdoxastic States', in C. Macdonald and G. Macdonald (eds), *Philosophy of Psychology: Debates on Psychological Explanation*, Volume One, Blackwell, Oxford UK and Cambridge USA, pp. 309–330.

Davies, M.: 1995b, 'Consciousness and the Varieties of Aboutness', in C. Macdonald and G. Macdonald (eds), *Philosophy of Psychology: Debates on Psychological Explanation*, *Volume One*, Blackwell, Oxford, UK and Cambridge, USA, pp. 356–392.

Hayes, P.: 1984, 'Liquids', in J. Hobbs (ed.), *Formal Theories of the Commonsense World*, Ablex, Hillsdale, NJ.

Horgan, T. and J. Tienson: 1988, 'Settling into a New Paradigm', Connectionism and the Philosophy of Mind: Proceedings of the 1987 Spindel Conference, *Southern Journal of Philosophy* **26**, 97–113, supplement.

Horgan, T. and J. Tienson: 1989, 'Representations Without Rules', *Philosophical Topics* **17**, 147–174.

Horgan, T. and J. Tienson: 1990, 'Soft Laws', *Midwest Studies in Philosophy: The Philosophy of the Human Sciences* **15**, 256–279.

Horgan, T. and J. Tienson: 1992, 'Cognitive systems as Dynamical Systems', *Topoi* **11**, 27–43.

Horgan, T. and J. Tienson: 1994, 'Representations Don't Need Rules: Reply to James Garson', *Mind and Language* **9**, 38–56.

Horgan, T. and J. Tienson: 1996, *Connectionism and the Philosophy of Psychology*, MIT Press, A Bradford Book, Cambridge, MA.

Horgan, T. and J. Tienson: 1999a, 'Short Précis of *Connectionism and the Philosophy of Psychology*', *Acta Analytica* **22**, 9–21.

Horgan, T. and J. Tienson: 1999b, 'Authors' Replies', *Acta Analytica* **22**, 275–287.

Marr, D.: 1982, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*, W. H. Freeman and Company, San Francisco.

Marr, D. and T. Poggio: 1976, 'Cooperative Computation of Stereo Disparity', *Science* **194**, 283–287.

Peacocke, C.: 1986, 'Explanation in Computational Psychology: Language, Perception and Level 1.5', *Mind and Language* **1**, 101–123.

Schiffer, S.: 1991, 'Does Mentalese have a Compositional Semantics?', in Barry Loewer and Georges Rey (eds), *Meaning in Mind: Fodor and his Critics*, Blackwell, Oxford UK and Cambridge USA, pp. 181–200.

Searle, J. R.: 1995, 'Consciousness, Explanatory Inversion and Cognitive Science', in C. Macdonald and G. Macdonald (eds), *Philosophy of Psychology: Debates on Psychological Explanation*, *Volume One*, Blackwell, Oxford, UK and Cambridge, USA, pp. 331–355.

Sejnowski, T. and C. Rosenberg: 1986, 'Parallel Networks that Learn to Pronounce English Text', *Complex Systems* **1**, 145–168.

Smolensky, P.: 1995, 'Reply: Constituent Structure and Explanation in an Integrated Connectionist/Symbolic Cognitive Architecture', in C. Macdonald and G. Macdonald (eds), *Connectionism: Debates on Psychological Explanation*, Blackwell Publishers, Cambridge USA and Oxford UK, pp. 223–290.

van Gelder, T.: 1991, 'What is the "P" in "PDP"? A Survey of the Concept of Distribution', in W. Ramsey, S. P. Stich, and D. E. Rumelhart (eds), *Philosophy and Connectionist Theory*, Lawrence Erlbaum Associates, Hillsdale, NJ and London, pp. 33–59.

Marcello Guarini
Humanities Department, Philosophy
The University of Michigan in Dearborn
4901 Evergreen
Dearborn, MI 48128
U.S.A.
E-mail: marcello_guarini@hotmail.com