Article

On Being Systematically Connectionist

LARS F. NIKLASSON **TIM VAN GELDER**

1 Introduction

In 1988 Fodor and Pylyshyn issued a challenge to the newly-popular connectionism: explain the systematicity of cognition without merely implementing a so-called classical architecture. Since that time quite a number of connectionist models have been put forward, either by their designers or by others, as in some measure demonstrating that the challenge can be met (e.g., Pollack, 1988, 1990; Smolensky, 1990; Chalmers, 1990; Niklasson and Sharkey, 1992; Brousse, 1993). Unfortunately, it has generally been unclear whether these models actually do have this implication (see, for instance, the extensive philosophical debate in Smolensky, 1988; Fodor and McLaughlin, 1990; van Gelder, 1990, 1991; McLaughlin, 1993a, 1993b; Clark, 1993). Indeed, we know of no major supporter of classical orthodoxy who has felt compelled, by connectionist models and arguments, to concede in print that connectionists have in fact delivered a non-classical explanation of systematicity.

Why has it been so unclear whether these models actually show that connectionism meets the challenge? Our view (apparently shared by Matthews, (forthcoming)) is that the most important reason has been obscurity in the concept of systematicity itself. In their 1988 paper Fodor and Pylyshyn discussed systematicity at length, but provided no succinct and precise characterization of it; at best, they gestured at the phenomenon with hints, analogies and anecdotal observations. They claimed that the systematicity argument in favor of the classical approach is a traditional one, but provided no references to previous occurrences, and as far as we can determine there is no occurrence of the argument or the concept in the cognitive science literature before 1988[1], except for the appendix of (Fodor, 1987). Consequently, despite the controversy created by the paper, there was simply no clear concept of systematicity available, and it was entirely unclear what kind of modeling, if any, could demonstrate systematicity.

In short, Fodor and Pylyshyn had set up a hurdle and challenged connectionists to jump it, but nobody knew quite where the top of the hurdle was. To make matters worse, subsequent attempts

---

Authors are listed in alphabetical order. We are indebted to Robert Hadley and Noel E. Sharkey for supplying useful discussions and comments on an early draft of this paper.

Address for correspondence: Lars F. Niklasson, Department of Computer Science, University of Skövde, S-54128, SWEDEN. Email: lars@ida.his.se. Tim van Gelder, Philosophy Program, Research School of Social Sciences, Australian National University, Canberra ACT 0200, AUSTRALIA. Email: tvg@coombs.anu.edu.au

by defenders of Fodor and Pylyshyn to clarify the concept of systematicity (e.g., McLaughlin, 1993a) not only failed to do so in any way that was of much practical help to connectionists, but to some extent shifted the ground. The hurdle was not only hard to see, it was moving as well.

Most connectionist models that were subsequently claimed to show that the systematicity challenge can be met did not have as their primary explanatory target the phenomenon of systematicity - and no wonder, since that phenomenon was so ill-defined to begin with! Consequently, suggestions that a particular model may have implications for the systematicity debate were usually made as an addition to the main arguments of the work, and typically did not involve detailed attention to exactly what systematicity is and what it would take for a connectionist model to explain it. It is thus not surprising that defenders of the classical approach remained unconvinced by connectionist claims to be explaining systematicity.

In this suffocating fog of vagueness and confusion, Robert Hadley's work has arrived like a gust of fresh air. Hadley has paid very close attention to exactly what systematicity is and what it would take for connectionist models to achieve it. He has given what is perhaps the first comprehensive, precise analysis of what a good number of the most well-known connectionist models in this area have really achieved in this regard. In particular, he has argued plausibly that we have no reason to believe that existing connectionist models have exhibited anything more than what he terms 'quasi-systematicity,' whereas humans exhibit at least 'strong systematicity'.

We have no quarrels with any of Hadley's major conclusions. Rather, we find Hadley's work a useful springboard into the continued investigation of human cognitive capacities and the power of connectionist models. In particular, Hadley is to be commended for clarifying systematicity to the point where *it is a relatively straightforward modeling exercise to demonstrate beyond any question that non-classical connectionist networks can exhibit an interesting form of systematicity.* In this commentary we will briefly describe one such model, which successfully handles Hadley's strong case.

Before proceeding, however, we wish to point out that it is by no means clear that *classical* architectures are capable of explaining the actual empirical facts of systematicity. There is no question that classical architectures exhibit systematicity of some form, and there is no question that human cognitive capacities exhibit systematicity of some form; *but are these forms the same*? Determining whether or not classical architectures explain the actual empirical facts requires (1) very close attention to psychological data to obtain a clear idea the exact way in which human cognitive capacities are systematic, and (2) careful comparison of the empirical facts with the kind of systematicity that is entailed by classical cognitive architectures. In (van Gelder and Niklasson, 1994) we argued that because steps (1) and (2) have not been carried out for most aspects of cognition, we do not now know that classical architectures do explain systematicity; further, in at least one area (a form of simple deductive inference) classical architectures manifestly *fail* to

1. The term 'systematicity' has appeared very occasionally, but when it did it meant something else entirely; see, e.g., Pylyshyn 1984 chapter 1. Note also that one component of the 1988 concept of systematicity, namely productivity, is of course very familiar; but a concept has not appeared merely because a component of it has, and Fodor and Pylyshyn explicitly declined to rely on productivity component.

explain the actual empirical facts. More generally, classical architectures appear to have a problem which might be expressed this way: while, as Hadley points out in section 2, connectionist architectures may be too sensitive to context to explain the empirical phenomena, classical architectures may not be sensitive enough, since the actual empirical facts do indicate some measure of what might be thought of as context-sensitivity. The upshot is that while it is an interesting open question whether connectionism can explain, in detail, the empirical facts of systematicity, it is also an interesting open question whether classical architectures can do so; further, any failure of connectionism in this regard does not automatically lend support to the classical competitor.

2 Kinds and Levels of Systematicity.

The moment Hadley attempts to render Fodor and Pylyshyn's vague, amorphous notion rigorous and precise, he finds it fractures into a number of kinds and levels. Thus, while the initial notion of systematicity was not tied to learning in any particular way - it asserted merely that cognitive capacities always come in 'clumps' (Fodor and McLaughlin, 1990) - Hadley found that to turn the systematicity challenge into a specific modeling problem for connectionists, it had to be reformulated as a learning problem: given that a system can acquire some capacities in a clump, does it automatically acquire other capacities in that clump? The precise nature of these clumps then becomes of paramount interest. Hadley precisely defines three 'levels' of systematicity in terms of the relationship between the capacities explicitly learnt by the network and the further capacities thereby acquired 'for free': weak, quasi- and strong systematicity. Additionally, at the end of his paper he describes informally a fourth level (or kind?), semantic systematicity.

In fact, there simply is no true, core notion of systematicity; one can precisely define a wide variety of kinds, levels and degrees of systematicity to suit one's theoretical purposes. For each notion of systematicity one defines, it then becomes an interesting theoretical question whether classical or connectionist architectures entail or at least are capable of exhibiting systematicity of that form, especially if that notion is closely tied to known empirical facts. While there is nothing intrinsically wrong with Hadley's selection of levels, we think there is an alternative set of levels which is more useful in the context of the learning capacities of connectionist models, a classification which is both simpler and more comprehensive.

We assume that the task before a connectionist network is to process sentences in some way or other that depends on the sentence's structure. Further, we assume that the network is trained on some subset of the relevant sentences, and can learn to handle every sentence in that subset perfectly; the network will exhibit systematicity if, in acquiring the ability to handle sentences in the training set, it also thereby automatically acquires the ability to handle *novel* sentences that are systematically structurally related to those in the training set. We then say that a connectionist network is systematic at level N if it is capable of successfully processing test sentences which are novel in the sense that:

Level 0. No novelty. Every test sentence appears in the training set.

Level 1. Novel Formulae. The test sentences themselves never appear in the training set, but all their atomic constituents appear in the same syntactic position somewhere in the training set.

Level 2. Novel Positions. The test sentences contain at least one atomic constituent appearing in some syntactic position in which it never appeared in the training set.

Level 3. Novel Constituents. The test sentences contain at least one atomic constituent which did not appear anywhere in the training set.

Level 4. Novel Complexity. The test sentences have a different level of complexity (embedding) than all sentences in the training set.

Level 5. Novel Constituents at Novel Complexity. The test sentences contain at least one novel constituent at a novel level of complexity.

Strictly speaking, there is no exact correspondence between any of these levels and Hadley's levels of weak, quasi- and strong systematicity. In particular, our level 3 ('NvG3') requires that the system is able to correctly process sentences containing some constituent that never appeared at all anywhere in the training set, whereas Hadley's level of strong systematicity ('HSS') requires that the constituent did appear somewhere in the training set, though not in the same syntactic position as in the test sentences. It is our belief that it is a more challenging task for a network to satisfy NvG3 than HSS, and this has been borne out in our modeling experience. Still, in order that there be no residual doubts at all, in what follows we describe a connectionist model that satisfies both NvG3 and HSS.

3 The Task

One sentence-processing task that obviously requires sensitivity to syntactic structure is transformation of formulae of propositional logic according to inference rules such as de Morgan's law. Any system that can perform such a task in a non-classical manner must do so without relying on the causal role of tokens of constituents in the tokens of the formulae being transformed (van Gelder, 1990). In other words, it must transform representations which have a compositional structure, but which are not constructed via concatenation. This has implications for both the choice of representation and the mode of composition. It would be trivial to have a connectionist network transforming, say, ’A & B’ into ’A v B’ if pools of 3 dedicated units (e.g., for ’p’, ’q’ and ’r’) had been used in positions ’A’ and ’B’, and a pool of 2 units for the connectives. The transformation is then reduced to a process that is sensitive to one constituent (i.e., the representation over the 2 units in the ’connective pool’) of the representation for the formulae. However, this approach suffers from two main problems; first of all it is open to the criticism of Fodor and Pylyshyn since it is nothing else than a concatenative mode of combination of the representations for the constituents, and secondly, as a result of the concatenative style of composition, it runs into problems with variable length representations, when embedded formulae are introduced.

To overcome the first problem, one can concentrate on the hidden layer representation in connectionist networks. This type of representation is a *distributed representation*, generated by non-concatenative composition of the combination of the representations presented at the input, and it has therefore no tokened constituents that could be used for structure sensitive operations. The model presented in the following is based on a special kind of connectionist architecture; a so called recursive architecture. This type of architecture allows that the distributed hidden layer representation for a combination of inputs (e.g., ’p’ and ’&’) could be copied back to the input and there be combined with representations for other constituents; atomic (e.g., ’q’) or complex (e.g., ’q v r’). This means that it also can overcome the second problem, mentioned above, i.e., generate finite length representations for complex expressions.

The heart of our model is a connectionist network (the Transformation Network, TN) which transforms distributed, non-concatenative representations of variable length logical formulae. It is this Transformation Network which is able to transform representations of formulae containing novel constituents with complete success.

The formulae to be transformed are constructed according to the following syntax:

A :: {p | q | r | s}

B :: {A | A v A | A -> A}

The transformation that the network is required to perform is the following:

A -> B   <==>   ~A v B

Thus, typical transformations the network is expected to perform are:

p -> q   <==>   ~p v q

p -> (q v r)   <==>   ~p v (q v r)

Now, the difference between HSS and NvG3 will mean that the training sets for the models, aiming to solve these levels, will differ. In both cases we will assume that the 's' is the symbol causing the novelty; in HSS it will not be allowed to appear to the left of the '->' or 'v' symbols (neither in simple nor embedded formulae). In NvG3 it will not be allowed anywhere in the training set. In this preliminary comparison, the differences meant that the training set for NvG3 contained:

(3 * 3) * 2   =   18

(3 * (3 * 2 * 3)) * 2   =   108

Total number of formulae: 126

while the training set for HSS contained:

(3 * 4) * 2   =   24

(3 * (3 * 2 * 4)) * 2   =   144

Total number of formulae: 168

and the total domain, also equalling the test set, contained 288 formulae.

After training, the two networks successfully processed all the formulae in the test sets. The exercise was repeated 5 times, with different sets of random initial weights, with the same successful result.

Since Hadley agrees (personal communication) that this modeling is indeed an example of HSS, but had some objections to the small set of symbols, an extended exercise was defined. The set of propositional symbols was extended with two symbols; giving a training set for NvG3 and HSS of 550 and 660 formulae, respectively, and a 936 formulae test set. After completed training the networks solved the task with full accuracy. Naturally, this second exercise required that the representational resources available to the network were increased (more on this in the next section). We see no reason in principle why this basic modeling approach cannot be extended to arbitrarily large domains.

4 The Model

In these exercises, distributed representations upon which the TN operated were formed by means of the Recursive Auto-Associative Memory (RAAM) architecture devised by Pollack (1988, 1990). The architecture is thus closely related to ones previously used by Chalmers, (1990); Chrisman, (1991); and Niklasson and Sharkey, (1992). The current model combines a RAAM and a Transformation Network (TN) into one architecture, shown in fig. 1.

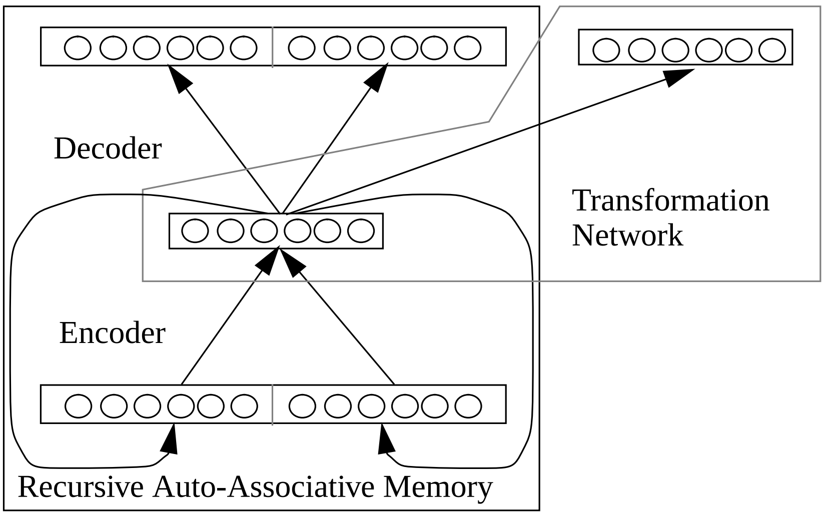

Fig. 1 The Combined Architecture

We will keep the description of this architecture to minimum; for a more detailed description see Niklasson, (1993); and Niklasson and van Gelder, (1994).

4.1 Training

Each pass through the training file of this combined architecture has two stages. First each formula in the training set is presented to the RAAM. This is a multi-step process for each individual formula. The network is first trained to encode/decode the combination of the representations for two atomic constituents, e.g. 'p' and '->'. The resulting activation on the hidden layer is then used as a distributed representation for this combination. This distributed representation is (at the input of the same RAAM) combined with representations for other atomic or complex constituents, e.g., 'q' or '(qvr)'. When all the constituents of a formula have been presented at the input, the activation on the hidden units in the last step of the process, is saved as a distributed representation for the actual formula, e.g. 'p->q' or 'p->(qvr)'. After each presentation of all the formulae to the RAAM, it is time for their presentations to the TN. The distributed representations for all the formulae are transformed from the hidden layer of the RAAM to the output of the TN.

Both parts of the architecture are trained using the backpropagation algorithm, and they co-evolve since both of them are trained on increasingly 'better' representations. Also, the distributed representations at the hidden layer of the RAAM (i.e. the representations for the complex constituents and complete formulae) are affected by both the encoding/decoding and the transformation processes, since the errors in both the RAAM and the TN are propagated to the Encoder.

4.2 Testing

After completed training (within about 4000 passes through the training set), the test set is presented. When the distributed representation for a formula is presented to the input of TN, there are basically two ways the results (i.e. the resulting output from the TN) can be evaluated; (i) compare the distributed representations generated at the output, with the distributed representations for all the formulae (and sub-parts of formulae) in the domain, by using Euclidean Distance in metric space, or (ii) present the output from the TN to the decoder part of the RAAM, and decode the representation for the complex expression back into its atomic constituents. This approach demands some way of separating a representation referring to an atomic constituent from one referring to a complex one.

Since this model is using distributed representations for both atomic and complex constituents, it has been extended with a facility that is (as a result of training) able to separate them automatically. This has been achieved by adding one unit to the representations for the constituents (activation 0 for atomic constituents and 1 for complex constituents). If the trained decoder signals an atomic constituent, the decoded representation is compared (by using ED) with the representations for all constituents (i.e., both atomic and complex) and complete formulae in the domain, and the closest one is chosen. If it, instead, signals a complex constituent, the representation is copied back to the hidden layer, for further decomposition. Since (ii) is harder to achieve, it is the approach we have chosen to use to evaluate our modeling exercises, but (i) is also evaluated simultaneously. When we, in the previous section, claimed success, the model correctly encoded, transformed and decoded (i.e., both (i) and (ii) above) all the constituents in the domain.

4.3 Representations

It would be impossible for a network to correctly process a sentence containing a novel constituent if it had no information at all concerning the syntactic category of that constituent. (This would be like somebody asking you to use the word 'pilk' correctly in English sentences without even telling you whether it is a noun or a verb.) How then do we ask the network to handle a novel constituent correctly, given its type, without training it on sentences involving that constituent? One way of handling this problem is to carefully choose, by hand, basic representations for the atomic constituents which reflect their syntactic category. Our solution to the problem is slightly different: we use a representation generator for atomic constituents, which encodes supplied type information in the form of tree structures, see fig. 2.

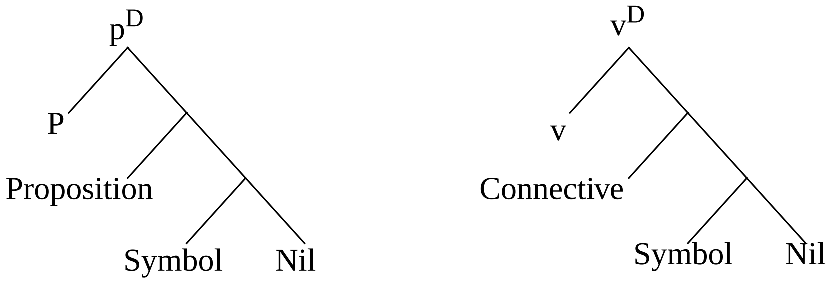

Fig. 2 Tree Structured Type Information

These tree structures were encoded (bottom-up) in a separate RAAM architecture (the representation generator), but one could also imagine an approach where an Elman-like architecture is used to extract that knowledge from syntactic context. All the leaves of the trees are assigned unique, non-overlapping, input representations, e.g.:

Table 1: Representations

Symbol	Representation
P	1 0 0 0 0
Q	0 1 0 0 0
R	0 0 1 0 0
S	0 0 0 1 0

The distributed representations for the atomic constituents are generated by training the separate RAAM to encode all these tree structures (except the one containing the novel constituent 's'), and then collecting the representations at the hidden layer. These distributed representations are then used in the training of the combined model, described above. In the first modeling exercise, with the smaller domain, each leaf, constituent and consequently complete formula (i.e the hidden layer of the RAAM in the combined architecture) was represented by 22 units, and in the second by 32 units.

Before testing the model, a distributed representation for the novel constituent has to be generated by the representation generator. Therefore, a unique non-overlapping representation is combined with the distributed representation for 'Proposition' at the input of the representation generator, and the hidden level activation is collected as a distributed representation for the novel constituent i.e. s^D = S + (Proposition + (Symbol + Nil)). This representation is then used in the combined model to generate the complete set of formulae, for testing.

It is important to note that though the combined architecture involves a number of stages, no part of the architecture is ever trained using any formula which contains the symbol 's' in any position, for NvG3, whereas it is allowed in some syntactic positions, for HSS.

5 Analysis

What evidence can we present, apart from the reported results from the modeling exercises, that would convince the doubtful reader that these networks do possess resources enabling systematic mappings from the representation for one formula to the representation for another? We believe that this is the final task that connectionists need to solve, in order to fully meet the systematicity challenge, i.e. not only present networks that exhibit systematically structure-sensitive processes, but also to explain why they exhibit that systematicity. Here we will here only give a simple explanation, involving the representations generated by the representation generator and used by the TN. The interested reader can also look at the explanations possible by the use of hyperplane analysis (e.g. Pratt and Kamm, 1991; Pratt, Mostow and Kamm, 1991; Sharkey and Jackson 1994).

The design and training regime of the representation generator results in representations that are systematically positioned in the space so that the representation for 's' occupies the space in between the 'known' constituents:

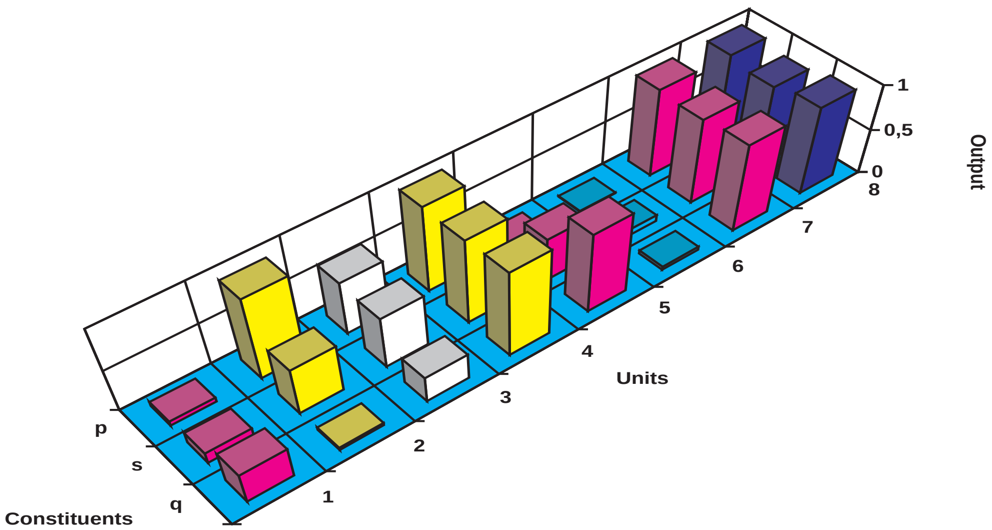

Fig. 3 Representations Formed by the Representation Generator

It is apparent, when investigating the representations used for the transformations, that (external) syntactically structure can be preserved by (internal) spatially structured representations, and that the spatial structure can be used for structure sensitive operations.

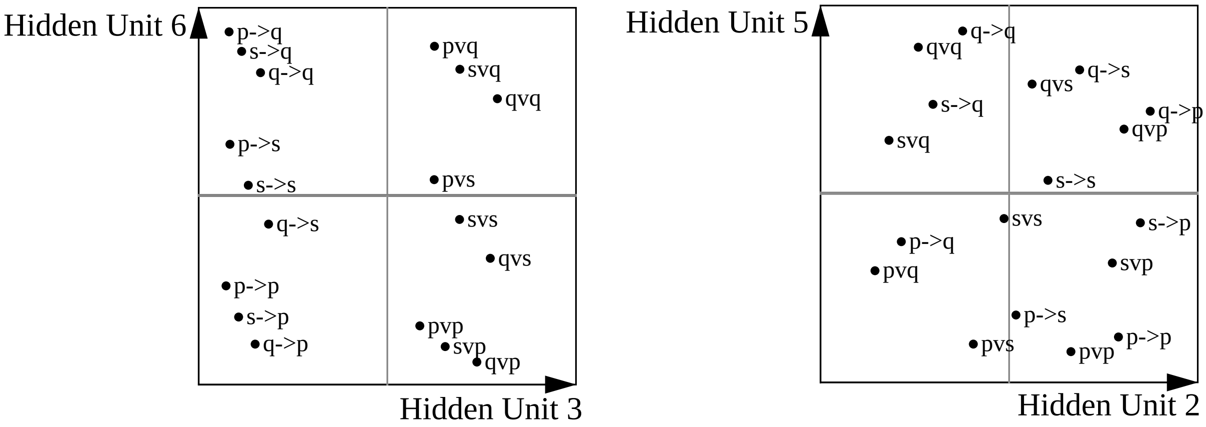

Fig. 4 Representations Used by the TN

Figure 4 shows the systematic structure of four dimensions of the representational space of a TN, but the same phenomenon could be noticed along all the dimensions (in this exercise, 8). It should be noted that these results were generated when the task (in order to reduce the number of dimensions) was reduced to a minimum:

Expressions:

A :: {p | q | s}

Transformation:

A -> A <==> A v A

Nevertheless, the figure clearly shows how operations sensitive to an external syntactical structure can be explained in terms of sensitivity to systematic spatial structure in the internal representational space of the network.

6 Conclusion

These exercises are not intended as psychologically realistic models of human cognitive processing. They possess many obviously implausible features, such as the use of backpropagation and exhaustive exposure to a training set. Our intention has not been to construct psychologically realistic models, but rather to demonstrate an in-principle point: that a certain interesting form of systematicity which is exhibited by humans is also exhibited by connectionist networks with the right architecture. Thus, some form of connectionist architectures may well subserve human cognition; systematicity phenomena, at least, do not rule this possibility out. Which of the classical or connectionist conceptions of basic architecture will eventually turn out to furnish the best models when all the relevant psychological data are taken into account remains an open question.

In our opinion the kinds of results described here, in the context of the kind of careful analysis of the concept of systematicity provided by Hadley, provide strong support for the general claim that systematicity considerations do not currently favor classical architectures over connectionist ones. We would like to conclude by stepping back to assess the impact of these points on the general plausibility of the classical conception of cognitive architecture. Over the years a variety of arguments have been advanced in support of the idea that the human cognitive architecture must be basically classical in form. It is interesting to ask why it is that in 1988, and in order to defend the classical conception against *connectionism*, Fodor and Pylyshyn came up with what, as far as we can tell, is an entirely novel argument. Presumably, it is because they themselves felt that the traditional arguments were no longer effective; they could not be used to make a compelling case for the classical conception as against connectionism. For example, one kind of traditional argument has been the 'universality' argument: that classical computational systems that are universal in the Turing sense are the only systems we know of that have the kind of flexibility to be adequate for modeling cognitive processes. Yet the obvious power and flexibility of connectionist architectures (and various 'universality' proofs associated with them) has effectively deprived this argument of all its persuasive force. A similar story applies to other traditional arguments such as the 'knowledge argument' (see, e.g., Pylyshyn, 1984, Ch. 1). The systematicity argument can therefore be seen a last ditch attempt to provide a decisive general argument in favor of the classical conception. Since the systematicity argument is undermined by the kind of results described here, it seems that there are no longer any powerful general arguments in favor of the classical view. This does not establish that the classical view is false or that the connectionist approach is right, but it does mean that it is only by providing the best detailed models of particular cognitive phenomena that either approach can claim empirical superiority. This is a contest which connectionism is ready, willing and able to enter.

7 References

Brousse O. 1993: Generativity and Systematicity in Neural Network Combinatorial Learning, Tech. Report CU-CS-676-93, Univ. of Colorado.

Chalmers D. J. 1990: Syntactic Transformation on Distributed Representations, *Connection Science*, **Vol. 2**, Nos 1 & 2, 53 - 62.

Chrisman L. 1991: Learning Recursive Distributed Representation for Holistic Computation, *Connection Science*, **Vol. 3**, No. 4, 345 - 366.

Clark A. 1993: *Associative Engines*: Bradford Books.

Elman J. L. 1990: Finding Structure in Time, *Cognitive Science* 14, 179 - 211.

Fodor J. 1987: *Psychosemantics: The Problem of Meaning in the Philosophy of Mind*: MIT Press.

Fodor J. A. and Pylyshyn Z. W. 1988: Connectionism and cognitive architecture: A critical analysis, In S. Pinker and M. Jacques (eds.), *Connections and symbols*: MIT Press, 3 - 71.

Fodor J. A. and McLaughlin B. P. 1990: Connectionism and the problem of systematicity: Why Smolensky's solution did not work, *Cognition*, **35**, 183 - 204.

Hadley R. F. 1992a: Compositionality and Systematicity in Connectionist Language Learning, *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society,* 659 - 664.

Hadley R. F. 1992b: Compositionality and Systematicity in Connectionist Language Learning, Tech. Report CSS-IS TR 92-03, Simon Fraser University, Burnaby, B.C., V5A 1S6, Canada.

Matthews R. F. Forthcoming: Three-Concept Monte: Explanation, Implementation, and Systematicity, *Synthese*.

McLaughlin B. P. 1993a: The Classicism/Connectionism Battle to Win Souls, *Philosophical Studies*, **70**, 45 - 72.

McLaughlin B. P. 1993b: Systematicity, Conceptual Truth, and Evolution, In C. Hookway and D. Peterson (eds.), *Philosophy and Cognitive Science*: Cambridge University Press, 217 - 234.

Niklasson L. F. and Sharkey N. E. 1992: Systematicity and Generalisation in Connectionist Compositional Representations, RR-92-01-005, University of Skövde, Sweden, Also forthcoming in G. Dorffner (ed.), *Neural Networks and a new 'AI'*: Chapman & Hall.

Niklasson L. F. 1993: Structure Sensitivity in Connectionist Models, In M. Mozer, et al. (eds.) *Proceedings of the 1993 Connectionist Models Summer School*: Lawrence Erlbaum, 162 - 169.

Niklasson L. F. and van Gelder T. 1994: Can Connectionist Models Exhibit Structure Sensitivity?, *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society.*

Pollack J. B. 1988: Recursive Auto-Associative Memory: Devising Compositional Distributed Representations, *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*, 33 - 39.

Pollack J. B. 1990: Recursive Distributed Representations, *Artificial Intelligence*, **46**, 77 - 105.

Pratt L. Y. and Kamm C. A. 1991: Improving a Phoneme Classification Neural Network Through Problem Decomposition, *Proceedings of the International Joint Conference on Neural Networks (IJCNN-91),* 821 - 826.

Pratt L.Y, Mostow J. and Kamm C. A. 1991: Direct Transfer of Learned Information Among Neural Networks, *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91),* 584 - 589.

Sharkey N. E. and Sharkey A. J. C 1993: Adaptive Generalisation, *Artificial Intelligence Review,* **Vol. 7**, No. 5, 313 - 328, Also In L. F. Niklasson and M. B. Bodén (eds.), *Connectionism in a Broad Perspective*: Ellis Horwood, 49 - 64.

Sharkey N. E. and Jackson S. A. 1994: Three horns of the representational trilemma, In Honavar and Uhr (eds.), *Symbol Processing and Connectionist Models for Artificial Intelligence and Cognitive Modelling: Steps Towards Integration*: Academic Press.

Sharkey N. E. and Jackson S. A. Forthcoming: An Internal Report for Connectionists, In R. Sun (ed.), *Computational Architectures Integrating Neural and Symbolic Processes*: Kluwer Academic Publishers

Smolensky P. 1988: On the proper treatment of connectionism, *The Behavioural and Brain Sciences*, **11**, 1 - 17.

Smolensky P. 1990: Tensor Product Variable Binding and the Representation of Symbolic Structures in Connectionist Systems, *Artificial Intelligence*, **46**, 159 - 216.

van Gelder T. 1990: Compositionality: A Connectionist Variation on a Classical Theme, *Cognitive Science*, **Vol. 14**, 355 - 364.

van Gelder T. 1991: Classical Questions, Radical Answers: Connectionism and the Structure of Mental Representations, In T. Horgan and J. Tienson (eds.), *Connectionism and the Philosophy of Mind*: Kluwer Academic Publishers.

van Gelder T. and Niklasson L. F. 1994: Classicalism and Cognitive Architecture, *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society.*