Can Connectionist Models Exhibit Non-Classical Structure Sensitivity?

Lars Niklasson[1]
Department of Computer Science
University of Skövde, S-54128, SWEDEN
lars@ida.his.se

Tim van Gelder
Philosophy Program, Research School of Social Sciences
Australian National University, Canberra ACT 0200,
AUSTRALIA
tvg@coombs.anu.edu.au

Abstract

Several connectionist models have been supplying non-classical explanations to the challenge of explaining systematicity, i.e., structure sensitive processes, without merely being implementations of classical architectures. However, lately the challenge has been extended to include learning related issues. It has been claimed that when these issues are taken into account, only a restricted form of systematicity could be claimed by the connectionist models put forward so far. In this paper we investigate this issue further, and supply a model and results that satisfies even the revised challenge.

Introduction

As is well known, in 1988, Fodor & Pylyshyn, arch-defenders of mainstream orthodoxy, threw down the mantle to connectionists, challenging them to explain the (so-called) systematicity of cognitive capacities without merely implementing a (so-called) classical cognitive architecture. Since then a number of connectionist models have been put forward, either by their authors or others, as in some measure either meeting the challenge, or suggesting that the challenge can be met in principle (for the models, see Pollack 1988, 1990; Smolensky 1990; Chalmers 1990; Niklasson & Sharkey 1993; Brousse 1993; etc.). Whether these models can or do meet the challenge has been the subject of much philosophical debate (Smolensky 1988; van Gelder 1990, 1991; Fodor & McLaughlin 1990; Sharkey & Jackson 1992; McLaughlin 1993a, 1993b; Clark 1993; Matthews 1994; etc.). A consensus has emerged that the only way to deliver a non-classical explanation of systematicity is to construct models that utilize representations that are compositionally structured, and hence can be the basis for structure sensitive operations, but are not constructed by strict concatenation, which is the hallmark of classical approaches. The model presented in this paper was constructed in accordance with this tradition.

In recent and lucid contributions to this debate Robert Hadley (1992, 1993) introduced a learning-based form of systematicity, and argued that a number of levels of systematicity, from weak to strong, ought to be distinguished, and that, while humans at least exhibit (what he defined as) strong systematicity, careful analysis shows that connectionist models have achieved at best (what he defined as) quasi systematicity.

In this paper, we first present an alternative hierarchy of levels of systematicity that is more simple and more general. We then present a connectionist model that can be trained to exhibit systematicity at both Hadley's strong level, and the (rather different) third level of our hierarchy, which is the capacity to deal appropriately with all sentences containing a totally novel constituent. This model demonstrates that the kind of implicit non-classical syntactic structure found in connectionist distributed representations is quite sufficient to support structure-sensitive operations, and to underwrite strong systematicity.

Note that this model is much too crude to be put forward as a serious model of real human capacities. It is intended as an exploration of the computational capacities of a certain kind of architecture, and to demonstrate that Fodor & Pylyshyn were wrong to argue that connectionism cannot, in principle, deliver a non-Classical explanation of systematicity.

What is Systematicity?

Insofar as the concept of systematicity differs from productivity, it was introduced into cognitive science for the first time in Fodor & Pylyshyn's 1988 paper. In view of this, it is a surprising fact that Fodor & Pylyshyn characterize systematicity only very vaguely. For example, Fodor has described systematicity as the idea that "cognitive capacities come in clumps" (Fodor & McLaughlin 1990, p 184). In this paper, we focus on just one component of the general phenomenon, the systematicity of inference. The 1988 paper gestured at this phenomenon as follows:

> ...organisms should exhibit similar cognitive capacities in respect of logically similar inferences... (p 47)
>
> You don't, for example, find minds that are prepared to infer John went to the store from John and Mary and Susan and Sally went to the store and John and Mary went to the store but not from John and Mary and Susan went to the store. (p 48)

Roughly, the idea is that any cognitive system that can perform one inference of a general type can perform other inferences of that type.

Unfortunately, the precise conceptual and empirical boundaries of the phenomenon of systematicity of infer-

1. Currently at University of Exeter, ENGLAND

ence in humans were left entirely unexplored in 1988, and indeed still are open issues. This gives rise to two problems. First, it is difficult to determine whether classical architectures can actually explain human systematicity of inference. This is because, while the compositional and structure-sensitive nature of classical architectures obviously provide a promising resource for explaining systematicity, it cannot be determined whether the nature of the systematicity actually entailed by the use of a classical architecture aligns sufficiently closely with empirically observable facts about humans (c.f. van Gelder & Niklasson 1994).

Second, and more relevant here, it was left entirely unclear what kind of modeling, if any, connectionists could do to convince Fodor & Pylyshyn et al. that their networks could deliver a non-classical explanation of systematicity of inference. The problem is that, while the phenomenon of systematicity of inference has no intrinsic connection with learning, connectionists are usually primarily interested in what kind of capacities a model can acquire on the basis of exposure to examples. The notion of systematicity should be related to the task that the network is trained to perform. So, how could the challenge of explaining systematicity be reformulated as a learning problem for connectionist networks?

Hadley focused on precisely this problem, and distinguished three levels of systematicity, weak, quasi- and strong. The relevant one for our purposes here is strong systematicity:

> We shall describe a system as strongly systematic if (i) it *can* exhibit weak systematicity, (ii) it can correctly process novel *simple* sentences and novel *embedded* sentences containing words in positions where they *do not appear* in the training corpus (i.e. the word within the novel sentence does not appear *in that same syntactic position* within any *simple or embedded* sentence in the training set). (1993 p. 6, his emphasis)

Note that Hadley stresses that this definition is intended to be read in such a way that the word within the novel sentence *does* appear in the training corpus, though not in the same syntactic position as in the novel sentence.

He then argued that none of the connectionist networks which seem to exhibit something like systematicity actually exhibit strong systematicity in this sense.

In our view, Hadley's three levels can be replaced with an alternative hierarchy that is both simpler and more comprehensive. (There is no question about which is the *right* or *true* hierarchy; it is just a matter of settling on a classification of levels which is the most clear and useful.) A trained network is systematic at level N if it is capable of successfully processing test sentences which are novel in the sense that:

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

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.

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.

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

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

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

Strictly speaking, Hadley's strong systematicity corresponds to none of our levels, since it is logically possible one could get a network that satisfied any one of our levels, and yet failed to exhibit systematicity at Hadley's strong level. However, it is our belief that satisfying our level 3 is more demanding than satisfying Hadley's level of strong systematicity, since it should be more difficult to get a network to handle an *utterly novel* constituent than to get it to handle one that already appeared in the training set, though not in the same syntactic position. This belief has been borne out in our modeling experiments.

The Task

The domain chosen here was simple inference in propositional logic. The aim is to show that a connectionist network that has learned to perform inferences of a certain type can thereby perform other instances of that type, including instances which contain an entirely novel constituent. The inference type is Material Conditional (Bonevac 1990):

A -> B => ~A v B

and the reverse.

Simple propositional symbols (p, q, r, s) were allowed for A, and simple propositional symbols, implications or disjunctions containing simple symbols (e.g. (p -> q)) were allowed for B. Thus, typical inferences the network is expected to perform are:

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

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

288 distinct inferences can be generated using the symbols (p, q, r, s). Of these, 162 contain at least one instance of the symbol *s*; 126 contain no instance of *s*. A network was trained to successfully perform all 126 inferences that contained no instance of the symbol *s*. This very same network was, without any further training, able to perform with 100% success the further 162 inferences that can be formed by using symbol *s*. The performance of this network was no mere statistical fluke; the same network architecture was trained 5 times using different randomly chosen starting weights with the same success rate on the test corpus. This result would satisfy our definition of systematicity at level 3.

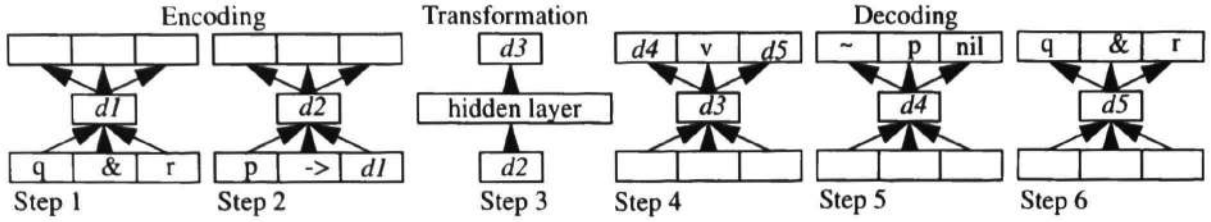

Figure 1: Holistic transformation of p -> (q & r) into ~p v (q & r)

In order to show that the network also is capable of exhibiting Hadley's strong systematicity, we made a change in the training set of the combined model. From the complete set of inferences, all those containing the symbol *s* to the left of a connective, both in simple (e.g., s -> q) and embedded formulae (e.g., p -> (s v q)) were removed, leaving us with a 168-formulae training set. The model was trained starting from the same five different sets of random weights as mentioned above. When tested on the complete set, this model also was 100% successful in handling the test corpus, in all of these five runs.

The Model

The model used here is an extension of the Recursive Auto-Associative Memory (RAAM) model devised by Pollack 1988 and used by Chalmers 1990. These models used two separate networks; one for encoding/decoding of representations for complex expressions (EN), and another for transformation of these representations (TN) as show in fig. 1.

The current model is inspired by Chrisman's (1991) architecture which combined the two separate parts of the models, mentioned above, into one architecture (as seen in fig. 2), for language translation.

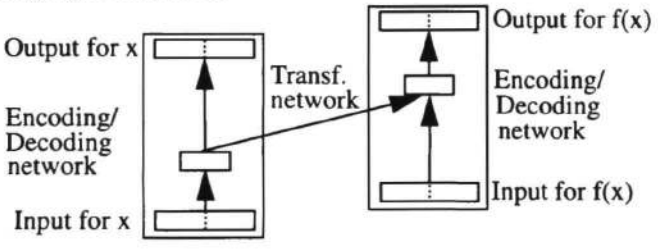

Figure 2: Chrisman's architecture

Our model uses only one dual RAAM as the EN and a TN directly connected to the hidden layer of that EN (see fig. 3). The EN has three layers of units, the input and output layers consist of $2(n+1)$ units (where n is the number of units chosen to represent an expression, complex or atomic) and the hidden layer consists of n units. Since the input layer of the TN is the same layer as the hidden layer of the EN, it, and the output layer of the TN, therefore consists of n units (for a more detailed description of the model, see Niklasson (1993).

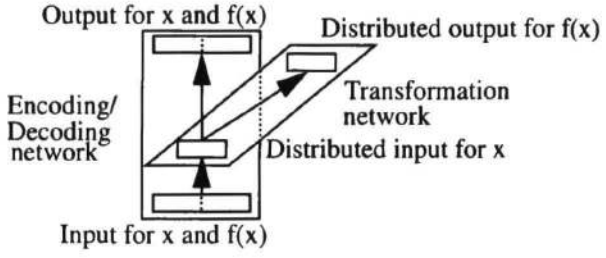

Figure 3: Our architecture

The important point to notice is that the two parts of the architecture co-evolve during the training phase, which means that the transformation part is trained on increasingly "better" representations (resulting from the proceeding training of the encoder/decoder). It is also important to notice that the input-hidden layer of the encoding/decoding network is updated as a result of the training of the transformation part, since its error is propagated to the encoder. This means that the hidden-layer representations are affected by both the encoding/decoding and the transformation processes. The simulations conducted, indicated that the model generalizes somewhat better with this feedback, than without it.

The Atomic Representation Generator

A connectionist network of the above general kind cannot be expected to successfully transform formulae/sentences containing a novel constituent if the representation for that constituent is utterly unlike anything with which it is already familiar. That would be like expecting a person to use a novel word (say *zork*) properly in sentences without knowing the slightest thing about the word itself, such as whether it is a noun or a verb. In our case, the network must at least somehow be able to tell that the novel constituent is an atomic proposition symbol.

To solve the problem of how to introduce a novel constituent while supplying to the network only the information that it is an atomic proposition symbol, we used a separate network to generate distributed representations for all atomic constituents. Several options for the generator are possible (e.g. the use of an Elman 1990, type of architecture, where the context of a syntax can be used). Several studies have shown that representations generated at the hidden layer of recurrent networks, are similar for

expressions with similar tree structures (e.g. Elman 1990, Pollack 1990). Here we use a the context of a class hierarchy to generate our distributed representations for atomic constituents in a RAAM (in a similar vein as Bodén & Narayanan, 1993).

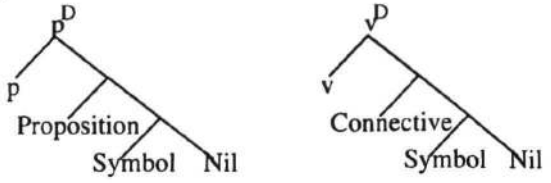

Figure 4: Context encoded by the representation generator

All the constituents in this domain (e.g. p, q, r, Proposition, Connective, Symbol, etc.) were assigned a unique, non-overlapping, representation of 22 units, of which two were active for each constituent (e.g. 1100000000000000000000 = p). A 44*22*44 RAAM was trained (with learning rate = 0.1, momentum = 0.1 for 40.000 iterations) to encode/decode these hierarchies. The distributed representations (e.g. p^D), i.e. the hidden layer representation, for the atomic constituents were then collected and used in the training of the combined architecture. Observe that the novel constituent, *s*, is not used in the training process.

Training and Testing of the Model

After training of the representation generator, the hierarchies are presented and the hidden-layer representation for them are collected as representations for the atomic constituents. These representations, are then used in the combined architecture to encode/decode and transform the formulae in the training set.

However, using distributed representations poses a new problem; how can it be decided (during the test phase) when a decoded representation refers to an atomic or a complex constituent? In models using pre-structured representations, it is possible to supply the decoder with information when to halt (e.g. the number of active units for an atomic constituent, as used by Chalmers). Here we adopt the technique of training the decoder, in the combined architecture, to automatically separate atomic from complex constituents. A single bit is added to the representations in order to differentiate atomic (0) from complex (1) constituents (c.f. Niklasson & Sharkey, 1993).

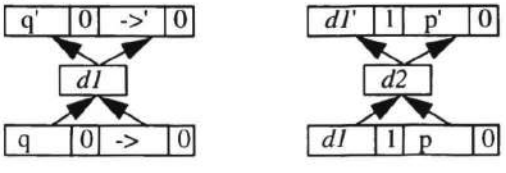

Figure 5: Automatic decoding

This explains why the combined architecture uses $2(n+1)$ input units.

The model is then trained for about 4000 iterations, using the standard backpropagation algorithm and a sigmoidal transfer function. For the encoder and the transformation network a learning rate of 0.02 was used, and for the decoder 0.05 was used. A momentum of 0.1 was used for all weights.

Before testing the model, a distributed representation for the novel constituent *s*, has to be generated. This is done by presenting a novel representation for *s*, e.g. 00000000000000000011, in combination with its context, i.e. the distributed representation for *'Proposition'*, to the representational generator (no additional learning takes place). The representation formed at the hidden layer (i.e. s^D) is collected and used to generate representations for the formulae, which are to be used in the combined architecture.

When the 288 formulae domain was presented as a test set, all the formulae were correctly encoded, transformed and decoded. In no case did the decoder incorrectly separate an atomic from a complex constituents (by using the last bit as an indicator) or identify the right atomic constituent incorrectly (by using Euclidean distance to the distributed representations for all expressions, i.e. both the representations for the atomic constituents, formed by the representational generator, and the representations for the formulae, formed at the hidden layer of the combined architecture).

Analysis of the Distributed Representations

In order to give an account why the model works, the task was reduced to transformation of only simple formulae, according to the following syntax:

A -> A <=> A v A

Only the propositional symbols p, q and s, were allowed for A and the negation was discarded, in order to reduce the number of dimensions in the representations.

The representation generator was trained to encode distributed representations for p, q, -> and v, by using the same type of class hierarchies discussed earlier. 8 units were used to represent each leaf:

P	10000000
Q	01000000
v	00100000
->	00010000
Proposition	00001000
Connective	00000100
Symbol	00000010
Nil	00000000

After the distributed representations had been collected, the combined architecture had been trained and the model correctly processed formulae containing the novel constituent *s* (after that its distributed representation had been generated by the representation generator, by combining 00000001 with Proposition^D), the hidden-layer representations formed in the RAAM were analyzed. The space available here prevents us from doing a thorough analysis, so we will resort to displaying the representations for the

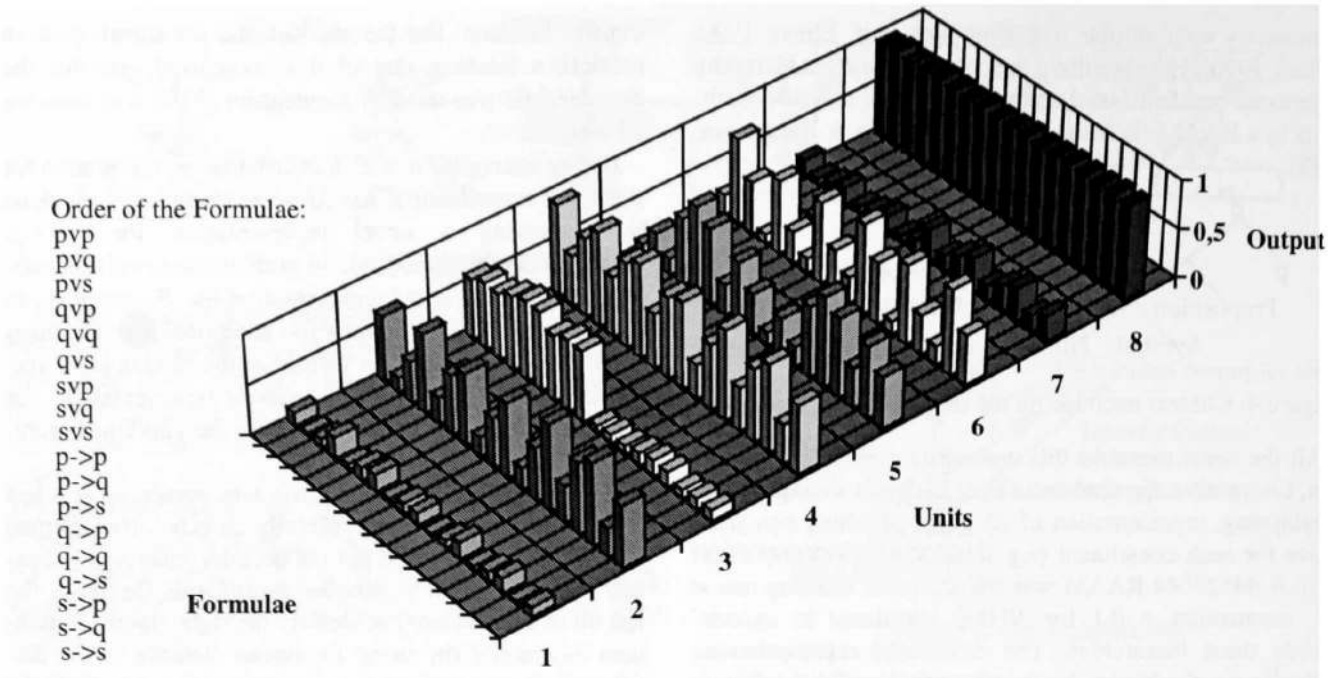

Figure 6: Hidden-layer representations

18 formulae in this domain, see fig. 6.

It is obvious that the result of the training is that the formulae are placed very systematically in the 8-dimensional space. We can use this spatial structure to explain how the network solves the transformation task. We need not even take the obvious choice of unit number 3, if we want to separate disjunctions from implications, as in fig 7.

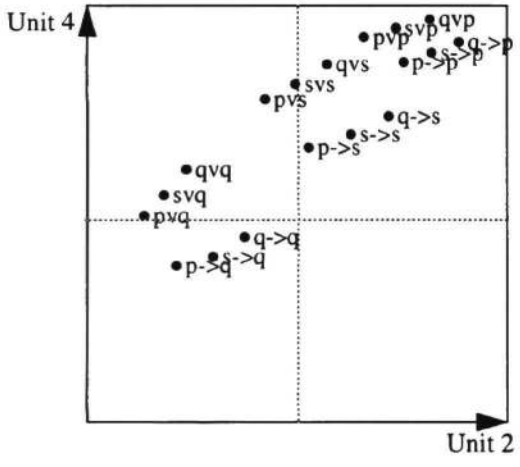

Fig. 7 Hidden unit space

It should be noted that this systematic spatial structure, and that the formulae with the novel constituent occupies the space between the known constituents, can be identified along all the dimensions.

Conclusion

This non-classical model achieves perfectly systematic performance at both our level 3 and at Hadley's strong level. We believe that this level of performance justifies the general claim that appropriately configured and trained connectionist networks exhibit systematicity i.e., that the Fodor & Pylyshyn challenge to connectionism has been unambiguously met. They would probably disagree and argue that we have missed the point, as Fodor & McLaughlin did in their reply to Smolensky:

> [the problem] is not to show that systematic capacities are *possible* given the assumptions of a connectionist architecture, but to explain how systematicity could be *necessary* how it could be a *law* that cognitive capacities are systematic - given those assumptions (p. 202, their emphasis)

But, as mentioned in the beginning of this paper, since the concept of systematicity, and the boundaries of the related clumps of cognitive capacities, are not unambiguously defined, it is, in our view, difficult to determine whether humans are necessarily systematic. This issue is, however, somewhat outside the scope of this paper. This issue aside, it should be noted that the model presented here is necessarily systematic (or at least shows some very strong evidence for assuming that it is), if systematicity is related to learning. Not only did the network exhibit the same behavior for the five successive runs, but if the hidden-layer representations are analyzed (which we unfortunately lack the space to do here in a more thorough fashion), very distinct mappings in the space can be identified.

Although Fodor & Pylyshyn did not explicitly relate systematicity to learning, they made some general remarks

about learning in connectionist models: "..these processes are all *frequency*-sensitive" (p 31 their emphasis). As we have shown, this is not true for all connectionist models, since the model presented here processes representations which it has not been explicitly trained on.

Note that the current model has not been shown to be capable of systematic performance at levels 4 and 5. We regard these levels of systematicity as important technical challenges for connectionists. Nevertheless, the current results demonstrate that there can no longer be any question about the "in principle" capacities of non-classical connectionist networks to exhibit systematic performance. The challenge now is to determine which approach to cognitive architecture is better able to describe and explain the fine detail of human capacities. However, in order to answer this question, much more critical attention must be paid to the concept of systematicity itself, and there must be much more empirical study of human capacities to ascertain the nature and limits of systematicity.

Acknowledgments

This research was, in part, supported by an award from the University of Skövde, Sweden to the first author and by an award from the Australian Research Council to the second author. We are indebted to Robert Hadley, Noel E. Sharkey and three anonymous reviewers for supplying useful discussions and comments on an early draft of this paper.

References

Bodén M. B. & Narayanan A., (1993). A Representational Architecture for Nonmonotonic Inheritance Structures. In *Proceedings of ICANN 1993*, Geilen & Kappen (Eds), Springer Verlag.

Bonevac D., (1990), *The Art and Science of Logic.*

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, pp 53 - 62.

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

Clark A., (1993), *Associative Engines*, Bradford Books.

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

Fodor J. A. & Pylyshyn Z. W., (1988), Connectionism a cognitive architecture: A critical analysis, In *Connections and symbols*, Pinker Steven & Mehler Jacques (Eds), MIT Press, pp 3 - 71.

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

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

Hadley R. F., (1993), Systematicity in Connectionist Language Learning, Tech. Report, 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**, pp. 45 - 72.

McLaughlin B. P., (1993b), Systematicity, Conceptual Truth, and Evolution, *Philosophy and Cognitive Science*, Hookway & Peterson (Eds), Cambridge University Press, pp 217 - 234.

Niklasson L. F. & Sharkey N. E., (1993), Systematicity and Generalisation in Connectionist Compositional Representations, (forthcoming) In *Neural Networks and a new 'AI'*, Dorffner G. (Ed), Chapman & Hall

Niklasson L. F., (1993), Structure Sensitivity in Connectionist Models, In Proc. of the 1993 Connectionist Models Summer School, Mozer M., et al. (Eds), Lawrence Erlbaum, pp 162 - 169.

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

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

Sharkey N. E. & Jackson S. A., (1992), Three Horns of the Representational Dilemma, In *Symbol Processing and Connectionist Models for AI and Cognition: Steps Towards Integration*, Honvar V. & Uhr l., (Eds), Academic Press.

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

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

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

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

van Gelder T. & Niklasson L., (1994), Can Classical Architectures Explain Systematicity, *The Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society.*