Connectionism and Systematicity

Steven Phillips

Department of Computer Science,
The University of Queensland, QLD 4072 Australia. Email: stevep@cs.uq.oz.au

Abstract

Linguists have known that adults (and perhaps even children) are *systematic* to a significant degree over the language domain. However, Connectionist models based on feedforward and recurrent architectures have failed to give a clear demonstration of *strong systematicity*. In this paper it is shown that these models, as they are used in the literature, cannot in fact demonstrate strong systematicity. Furthermore, it is argued that the critical issue in demonstrating systematicity is not one of *concatenative/functional* compositionality, but one of the relationship between *constructive* and *deconstructive* processes.

Introduction

The consistent representation and process of complex objects, constructed from the combination of simpler objects, is termed, Systematicity [5]. How is it that adults behave systematically over a vast (combinatorial) number of sentences and concepts from relatively few examples?

Connectionism attempts to explain systematicity through a learning mechanism embedded in some suitable environment. However, the major problem facing the Connectionist approach is determining an *appropriate* learning mechanism: one that is sufficiently unbiased to allow the acquisition of a range of behaviours, yet sufficiently robust to acquire a particular behaviour from a variety of possible training sets. This problem is well known as the bias/variance dilemma [6].

Although Connectionist models have sufficient variance to approximate *most* functions/behaviours with arbitrary accuracy [9] [15], the important question, is: do the models have sufficient bias to account for systematicity?

Previous answers to this question have been an emphatic *No*, based on philosophical arguments on the nature of internal representations and processes [5] [4], and statistical analyses of training sets [8]. This paper examines this question from a computational perspective.

1 Systematicity

Systematicity is the ability to represent (*systematicity of representation*) and infer (*systematicity of inference*) structurally related objects [5].

Consider, for example, a domain of objects related by the underlying structure: $Subject \times Verb \times Object$, where $Subject$ is the set of unstructured objects $\{Bill, Mary, John\}$, $Verb$ is the set $\{loves\}$, $Object$ is the set $\{Bill, Mary, John\}$, and $\times$ is the cartesian product operator generating all in-order combinations, or 3-tuples (e.g., *Mary loves Bill* and *John loves Mary*).

The degree by which a model is systematic of representation can be capture by its generalization performance on auto-association tasks. Within a Connectionist framework, auto-association requires the induction of two vector functions f and g such that, $f(s) = \mathcal{R}[s] \wedge g(\mathcal{R}[s]) = s$, where f maps an external representation of the object $s \in \cap D$ (some structured domain) to an internal representation $\mathcal{R}[s]$, and g performs the reverse mapping.

Similarly, within the Subject-Verb-Object domain for example, systematicity of inference can be measured by generalization performance on the mapping: $f(s, \mathcal{Q}_i) = x_i$, where $\mathcal{Q}_i$ is the question vector requesting the i *th* component x_i. For example, given the sentence *John loves Mary*, and the question *Who loves ?*, the resulting component is *John*. Clearly, this task is very simple, but it contains sufficient structure to demonstrate the deficiencies in Connectionist models of structured domains. Systematicity of inference can be considered a more general case of systematicity of representation where some component of the complex object is recovered rather than the whole object.

Subsequent to Fodor and Pylyshyn's work, Hadley [8] defined three degrees of systematicity:

- *Weak systematicity* - generalization from a training set contained each possible component appeared in each possible position, though not necessarily each possible combination of components;
- *Quasi-systematicity* - an extension of weak systematicity to sentences with embedded sentences

where the enclosed and embedded sentences are structurally isomorphic; and

- *Strong systematicity* - an extension of weak systematicity where it is not necessary for the training set to contain components in all possible positions.

For example, a model that has only seen *Mary* in the *Subject* position, somewhere in the training set, and that generalizes to cases where *Mary* is in the *Object* position, is strongly systematic. Hadley's strong systematicity applies equally to systematicity of representation and systematicity of inference, and clearly, Fodor and Pylyshyn intended that cognitive systems were at least strongly systematic in Hadley's sense.

Quasi-systematicity is relevant to recursive structures so need not be of concern here. (See also [12] for further definitions.) The important point is that Connectionist models have not demonstrated strong systematicity [8]. A statistical analysis of the training sets for models based on the recurrent network architecture, showed that in all likelihood each component would have appeared in each position. Consequently, there has not been a clear demonstration of strong systematicity.

In the next two sections, the feedforward and recurrent networks are examined to determine whether they *can* in fact demonstrate strong systematicity with respect to representation and inference.

2 Systematicity of representation

Previous studies demonstrating structured processing by feedforward networks showed a very high degree of generalization on the auto-association of N-tuples [2] [13]. Of importance here, is whether the degree of generalization qualifies as strong systematicity.

The input/output representation of tuples in [2] was such that one input unit for each component in each position indicated the component's presence or absence in that position of the tuple. For the Subject-Verb-Object domain this scheme requires 3 + 1 + 3 (totaling 7) input units to represent all possible tuples. At the output layer, each hyperplane must separate tuples represented in hidden unit activation space into two groups: those tuples that contain the corresponding component-position (e.g., *John* in *Subject* position), and those where this combination is absent.

To demonstrate strong systematicity there must be at least one component in one of the positions (e.g., *John* in *Subject* position) that does not appear in the training set. With respect to the *John-Subject* hyperplane, in this case, there only exists one type of point in the training set (i.e., *John* not in *Subject* position). Consequently, the training set provides no information to discriminate between the presence and absence of *John* in the *Subject* position. Since the orientation of this hyperplane is independent of all other hyperplanes, the network cannot be expected to generalize to tuples with the *John-Subject* component. This result, of course, applies regardless of the number of possible values in the subject (or object) position. Thus, the feedforward architecture (as used in [2]) cannot demonstrate strong systematicity.

In [13], the tuples at the output layer were represented using a *block* encoding scheme [1]. That is, within each group of output units associated with a given position, 50% were *on* consecutively, with wraparound, and the rest *off*. Components differed from each other by a right-shift operation on the representation. This encoding scheme, however, introduces *a priori* an ordering over components [1]. In which case, strong systematicity may be possible, but as a consequence of word level structure (similarity). Systematicity, however, is a property at the sentence level of structure, not the word level [5]. Therefore this work cannot be considered as a valid demonstration of strong systematicity.

The lack of strong systematicity is because the weights that map the subject component from the input to the hidden, and from the hidden to the output are independent of the weights that perform a similar mapping for the object component.

Dependency can be introduced by tying these weights so that the update of one weight enforces an update of its tied weight. This technique has been used to introduce translational invariance for optical character recognition. An alternative approach is to use a recurrent network where components are presented one per time step. Thus dependency is a natural consequence of the architecture. Although either approach may address systematicity of representation, in the next section it is shown that they cannot address systematicity of inference.

3 Systematicity of inference

Recurrent networks, used to demonstrate generalization over structured domains (e.g., the architectures of Elman [3], Jordan [10], and Pollack [14]), all share a common feature. That is, they all additively combine some non-linear transformation of representations at the input and context layers. It is argued that this feature prevents these architectures from demonstrating strong systematicity.

The argument proceeds by first describing an inference task, and then showing that the network cannot represent a solution to this task with a single hyperplane per component at the hidden layer. Consequently, it is argued that the network cannot be strongly systematic.

The task used is a simplified version of the inference task described in section 1, so that there are just two possible components x and y, and one implicit binary relation (i.e., the name of which is not actually presented to the network). The simplified task is to acquire the mapping:

$$f(\mathcal{R}[(x,y)], \mathcal{Q}_1) \rightarrow x$$
$$f(\mathcal{R}[(x,y)], \mathcal{Q}_2) \rightarrow y$$
$$f(\mathcal{R}[(y,x)], \mathcal{Q}_1) \rightarrow y$$
$$f(\mathcal{R}[(y,x)], \mathcal{Q}_2) \rightarrow x$$

where $\mathcal{R}[(x,y)]$ is a representation of the binary relation (x,y), and $\mathcal{Q}_1$ and $\mathcal{Q}_2$ are the question vectors, requesting the first and second arguments respectively, and f is the function that implements the mapping.

In order to recover one of the components, say x, by a single hyperplane at the hidden layer, the hyperplane must be positioned so as to satisfy the following equations:

$$W_I.\vec{\mathcal{Q}}_1 + W_C.\vec{\mathcal{R}}^C_{x,y} + \vec{B} > 0 \qquad (1)$$
$$W_I.\vec{\mathcal{Q}}_2 + W_C.\vec{\mathcal{R}}^C_{x,y} + \vec{B} < 0 \qquad (2)$$
$$W_I.\vec{\mathcal{Q}}_1 + W_C.\vec{\mathcal{R}}^C_{y,x} + \vec{B} > 0 \qquad (3)$$
$$W_I.\vec{\mathcal{Q}}_2 + W_C.\vec{\mathcal{R}}^C_{y,x} + \vec{B} < 0 \qquad (4)$$

where $\vec{\mathcal{R}}^C_{x,y}$ is the vector representation at the context layer of the binary relation (x,y). $\mathcal{Q}_1$ and $\mathcal{Q}_2$ are the representations of the question vectors at the input layer. Subtracting equation (2) from equation (1), and equation (4) from equation (3) leaves:

$$W_I.(\vec{\mathcal{Q}}_1 - \vec{\mathcal{Q}}_2) > 0 \qquad (5)$$
$$W_I.(\vec{\mathcal{Q}}_2 - \vec{\mathcal{Q}}_1) > 0 \qquad (6)$$

Since there does not exist a weight matrix which satisfies both equations (5) and (6), the network cannot represent the solution with a single hyperplane for each component. Consequently, there requires at least two hyperplanes per component at the hidden layer, where one plane discriminates between an x in the first position and all other points, and a second plane discriminates between an x in the second position and all other points. A third hyperplane at the next layer, combines the results of the two hyperplanes at the hidden layer. The positioning of the third hyperplane is dependent on the positioning of the hyperplanes at the hidden layer, which in turn, can only be positioned correctly when each of these planes has *seen* at least one point from each of the two classes they discriminate. That means, at least one point where x is in the first position, and one point where x is in the second position. This reasoning applies so long as there are at least two possible components in each position. Therefore, the network cannot demonstrate strong systematicity.

This result is fairly general in that it holds regardless of the representation of the question vectors, or the binary relation, and whatever (non)linear transformations may be applied to the representations before they are additively combined. Consequently, the result also applies to the feedforward network.

The limitation of the result is that it applies to networks where information from the input and context spaces is brought together primarily under addition, and extracted by hyperplanes. It does not apply for example, to networks where representations from the input and context are multiplied (e.g., [7]). However, in the next section, it is argued that it is not the mode combination, nor extraction, in itself, that is the critical issue, but the relationship between these two functions.

4 Structure sensitivity

Since van Gelder's [17] distinction between two types of compositionality: *concatenative* and *functional*, Connectionists have believed that Fodor and Pylyshyn's [5] contention (of structure sensitivity) can be circumvented by functional compositionality. However, in what follows is an outline of an argument suggesting that the functional/concatenative distinction is a tangential issue.

Consider two constructive functions, **concat** (whose mode of construction is concatenation) and **multi** (whose mode of construction is multiplication), for example:
concat (JOHN, ELSON) → JOHNELSON
multi (3, 4) → 12.
Now consider the two associated deconstructive functions, **decat** and **factor** which must extract the components from the compositional representation. In both cases, the decomposition is ambiguous. **JOHNELSON** could be decomposed into **JOHN** and **ELSON**, or **JOH** and **NELSON**. Possible decompositions of **12** are (**3**, **4**), (**11**, **1**), or even (**1**, **2**).

The ambiguity in determining the components of a complex representation arises because the *explicitness* of a component within a representation is relative to the deconstructive process [11]. The representation **12**, in itself, does not restrict the possible decompositions. This illustration is effectively Fodor and McLauglin's [4] point as to why Smolensky's [16] tensor representational system, in itself, does not guarantee structure sensitivity. The deconstruction process must *know* about the mode of construction.

However, in addition to this point, a constructive function (such as **concat** and **multi**) must map each distinct pair to a unique represention. This condition can be achieved with **concat** by requiring a space between the two components, and with **multi** by multiplying the second component by a sufficiently large factor. However, the point is that there are an infinite number of such schemes, and therefore an infinite number of possible deconstruction functions. To ensure systematicity, the class of such functions must be restricted, or in other words, the network must be biased so as to enforce this restriction.

Clearly, appropriate biases are contingent on the modes of construction. However, the results of this paper suggest that merely making available a sufficiently expressive function class does not guarantee such structural properties as systematicity. An important step in the development of more powerful Connectionist models is the characterization of biases that enforce such properties.

Acknowledgements

The author would like to thank Janet Wiles for comments on a previous draft of this paper, and Simon Dennis and Cyril Latimer for discussions on some related issues. This work was supported by a University of Queensland Postgraduate Research Award.

References

[1] P Bakker, S Phillips, and J Wiles. The N-2-N encoder: A matter of representation. In S Gielen and B Kappen, editors, *ICANN'93: Proceedings of the International Conference on Artificial Neural Networks*, pages 554–557, London, September 1993. Springer-Verlag.

[2] O J Brousse and P Smolensky. Virtual memories and massive generalization in connectionist combinatorial learning. In *Proceedings of the 11th Annual Conference of the Cognitive Science Society*, pages 380–387, Hillsdale, NJ, 1989. Lawrence Erlbaum.

[3] J L Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 1990.

[4] J A Fodor and B P McLaughlin. Connectionism and the problem of systematicity: Why Smolensky's solution doesn't work. *Cognition*, 35:183–204, 1990.

[5] J A Fodor and Z W Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28:3–71, 1988.

[6] S Geman, E Bienenstock, and R Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.

[7] C L Giles, G Z Sun, H H Chen, Y C Lee, and D Chen. Higher order recurrent networks and grammatical inference. In D S Touretzky, editor, *Advances in Neural Information Processing Sytems 2*, pages 300–307. Morgan Kaufmann, 1990.

[8] R F Hadley. Compositionality and systematicity in connectionist language learning. Technical Report CSS-IS TR 93-01, Simon Fraser University, Burnaby, BC, 1993.

[9] K Hornik, M Stinchcombe, and H White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[10] M I Jordan. Serial order: A parallel distributed processing approach. Technical report, MIT, Cambridge, MA, 1990.

[11] D Kirsh. When is information explicitly represented? In P Hanson, editor, *Information, Language and Cognition: Vancouver Studies in Cognitive Science*, pages 340–365. UBC Press, Vancouver, BC, 1991.

[12] L Niklasson. Structure sensitivity in connectionist models. Technical Report RR-93-01-010, University of Skovde, 1993.

[13] S Phillips and J Wiles. Exponential generalizations from a polynomial number of examples in a combinatorial domain. In *Proceedings of the International Joint Conference on Neural Networks*, pages 505–508, Nagoya, Japan, 1993.

[14] J B Pollack. Recursive distributed representations. *Artificial Intelligence*, 46:77–105, 1990.

[15] M Sato, Y Murakami, and K Joe. Learning chaotic dynamics by recurrent neural networks. In *Proceedings of the International Conference on Fuzzy Logic and Neural Networks*, 1990.

[16] P Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46:159–216, 1990.

[17] T van Gelder. Compositionality: A connectionist variation on a classical theme. *Cognitive Science*, 14:355–384, 1990.