Content, Context and Connectionist Networks

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

Mikael Bodén
Department of CS and EE, University of Queensland,
4072, AUSTRALIA/
Department of Computer Science
University of Skövde, S-54128, SWEDEN
mikael@ida.his.se

Abstract

The question whether connectionism offers a new way of looking at the cognitive architecture, or if its main contribution is as an implementational account of the classical (symbol) view, has been extensively debated for the last decade. Of special interest in this debate has been to achieve tasks which easily can be explained within the symbolic framework, i.e., tasks which seemingly require the possession of a systematicity of representation and process, in a novel way in connectionist systems. In this paper we argue that connectionism can offer a new explanational framework for aspects of cognition. Specifically, we argue that connectionism can offer new notions of compositionality, content and context-dependence based on connectionist primitives, i.e., architectures, learning, weights and internal activations, which open up for new variations of systematicity.

Introduction

Ever since Fodor and Pylyshyn (1988) published their seminal paper in which they defined the relation between systematicity (i.e., the systematic structure of mental representations and the structure-sensitivity of mental processes), compositionality (i.e., the method of composing/ decomposing structured mental representations) and the cognitive architecture, the debate has been intense. It has had two main research agendas; i) to exhibit and explain systematicity in connectionist systems (Smolensky, 1990; van Gelder, 1990; Pollack, 1990; Chalmers, 1990; Niklasson and Sharkey, 1992, Niklasson van Gelder, 1994, Phillips, 1994) and ii) to question the relevance of the systematicity and compositionality phenomenon altogether (Goschke and Koppelberg, 1991; van Gelder and Niklasson, 1994; Matthews, 1994).

The success of the early connectionist counter examples was questioned by Hadley (1992, 1994a). He noted that in many of these examples the success might have been due to the constitution of the training set. He therefore re-formulated systematicity in a learning-based fashion, defining different levels of systematicity depending on the content of the training set. He identified three levels of systematicity:

- weak systematicity (concerned with generalization to novel sentences in which tokens appear in syntactic positions in which they have appeared during training),
- quasi systematicity (requires weak systematicity and embedded structures),
- strong systematicity (requires quasi systematicity and generalization across syntactic positioning of tokens).

Hadley argued that no counter-example had achieved the strongest form of systematicity, with the possible exception of Niklasson and van Gelder (1994). Hadley was, however, concerned about the approach adopted by Niklasson and van Gelder for generating representations. They used a separate network which encoded syntactic information in order to generate similar distributed representations (i.e., close in the representational space) for tokens of similar types, which caused Hadley (1994b) to classify their result as a 'border line' case.

Recently, Phillips (1998) pointed out that connectionist architectures using localistic representations, by themselves cannot account for strong systematicity. But also, that this restriction does not preclude separate mechanisms for generating similarity based representations, which could be used in subsequent systematicity tasks. He outlined two research directions; develop architectures which could support systematicity under localistic input/output representations, or justify similarity-based distributed representations sufficient for allowing systematicity.

The former of these directions is exemplified by Hadley and Hayward (1997) when they showed that a network could achieve an even stronger form of systematicity; semantic systematicity, defined as:

> A system possesses semantic systematicity if it is strongly systematic and it assigns appropriate meanings to all words occurring in novel test sentences which (would or could) demonstrate strong systematicity of the network (Hadley, 1994b, p. 434).

The intention of this paper is to take the latter research direction pointed out by Phillips (1998), i.e., to justify similarity-based representations sufficient for systematicity. Two forms of justifications can be identified; i) empirical justification of the exact boundaries of the systematicity phenomenon (an analytic approach), or ii) a technical justification related to the representational primitives of connectionist architectures (a synthetic approach).

We will, in the following, take the latter of these and define meaning (i.e., content) in relation to connectionist

architectures, learning, weights and internal activations, and show that the implications of this approach are rather different compared to the notions within classicism. We argue that our view allows systematic processes which are sensitive not to the syntax of the representation (which is a cornerstone of the traditional definition) but instead to the *context* in which the *content* of the representations is defined. This context could naturally also include processing of expressions based on their syntactic structure.

To substantiate our arguments, we will present some examples and performance results which assign the appropriate meaning (admittedly, somewhat different than defined by Hadley) to novel test cases. The examples can also be used to indicate how our approach can be empirically validated or refuted.

If our approach is accepted, that would allow connectionists to go beyond traditional symbol processing and account for context-dependent semantic systematicity.

Content of connectionist representations

In a classical system appropriate meaning can be assigned to words depending the structural positioning of their representational tokens. This is due to the definition of a classical system which hinges on the possession of a combinatorial syntax and semantics for mental expressions. The definition states that the content of a complex representation is a function of the meaning of the constituents, together with the constituent structure of the representation.

The kind of connectionist system we have in mind, does not possess *syntactically* structured representations. Instead it relies on the possession of *spatially* structured representations, formed as a result of an individual learning situation.

The main difference between the two approaches is what can be assumed when trying to extract content from the representations. In a connectionist system, the spatial structure is the result of a specific learning situation. It is generally not, contrary to the classical approach, possible to assume a surrounding context when constructing representations.

In order to objectively compare the two approaches we argue that they must be allowed to make the same assumptions about the context when constructing systematicity examples. Rather than using natural language examples (where it is difficult to define the complete contextual framework), we will here use a different domain including reasoning with both defaults and exceptions.

We propose the following understanding of content and context within the connectionist framework:

- The only *context* supplied to a network is defined in terms of its training set. Any *content* found in the representations depends on this context.
- The organization of the representations is not arbitrary, rather it depends on the *context* expressed in the training set.
- The weights operating on the representations extract the *content* expressed in them. Therefore, the weights and the internal dynamics of the receiving units can be used to define *content* which entails that an explanation to systematic processes working on the representations, is possible.

One possible objection to this view could be the definition of content, but we argue that it is in line with Palmer's (1978) definition of information in cognitive representation:

> The only information contained in a representation is that for which operations are defined to obtain it (Palmer, 1978, p. 266)

Figure 1 exemplifies some of the above points. Two input units (x and y) are connected to a logistic output unit (z) with weights -3.7 and 9.4 respectively. In addition to this, a -3.2 bias weight is connected to the output unit. The figure shows how a particular weight configuration partitions the input space, allowing the extraction of the content of three sample input (A, B and C in Figure 1) representations to the network.

The context (i.e., the relations expressed in the training set) in this example is that representations B and C belong to the same category (z=0), and A to a different one (z=1). As can be seen in the figure, the network has in this particular case learned this classification. It is also clear that we now can use the weights in the trained network to extract the content in the input representations, even for novel ones, and identify spatial regions for the different classes.

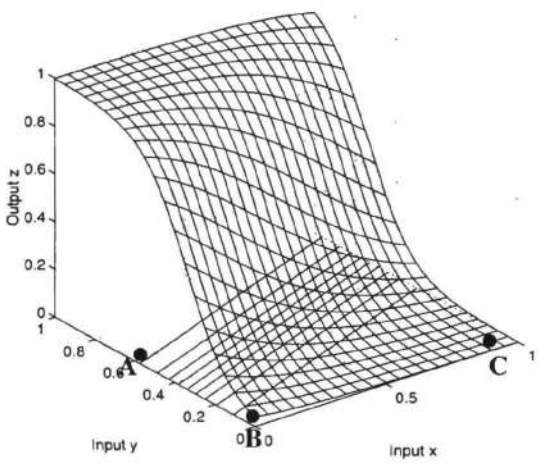

Figure 1: The result of a particular learning situation. The lines represent different values for z (0.9 to 0.1)

What this simple example does not show is how the organization of the representations (i.e., the locations to points in the n-dimensional representational space) can be made sensitive to the particular context expressed in the training set. For this, we need to extend the architecture with features allowing learnable representations for tokens. Here

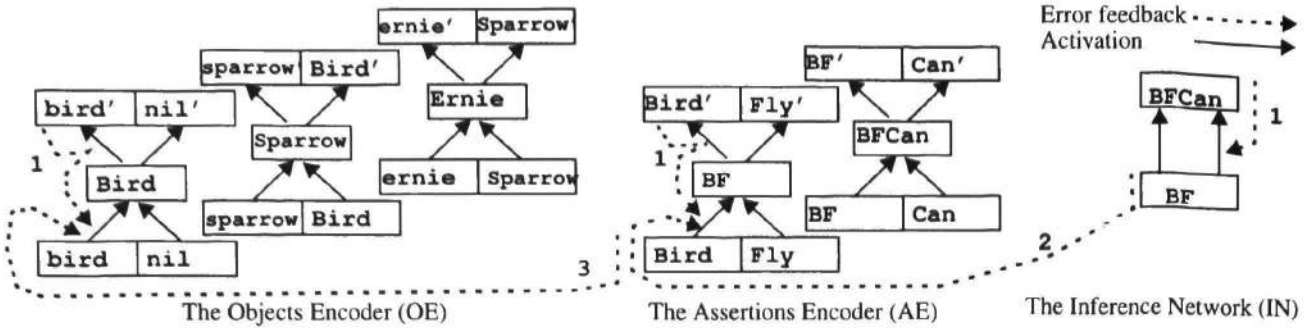

Figure 2: The complete architecture. The OE and the AE are standard RAAM networks. It should be noted that the figure show the same network but at different stages of the encoding process. Numbers indicate different error feedback; 1 is the feedback within a network, 2 is feedback between the IN and the AE and 3 is feedback from the AE to the OE. The between-network feedback is possible since a separate dictionary is used to store all representations (e.g., **Bird**) and the relation to its constituents (e.g., **bird** and **nil**).

we use the same architecture as used by Bodén and Niklasson (1999), see Figure 2. This architecture is intended to incorporate different kinds of context. Here we will use a simple hierarchical taxonomy (e.g., that **Sparrows** are **Birds** and that **Ernie** is a **Sparrow**) and, in the Objects Encoder (OE), generate compositional representations based on this context. In addition to this, the representations needed to train and test the Inference Network (IN) need to be generated (i.e., (**Bird Fly**), ((**Bird Fly**) **Can**)). This is done by the Assertions Encoder (AE). Finally the particular inferences valid in a particular context (e.g., that birds in fact can fly) are trained in the IN. The encoders are standard RAAM networks (Pollack, 1990) and the inference network is related to Chalmers' (1990) transformation network. The main difference from Chalmers is the use of between-network error feedback (which is an extension to Chrisman's (1991) confluent representations).

The within- and between-network error feedback (see Figure 2) allows that the representation for an object (e.g., **Ernie**) is affected by its relation to other objects in the domain, the assertions it appears in and the valid inferences it is part of. It will therefore in the following be referred to as contextual feedback.

An illustrative example

The OE was trained to encode the following:

Node:	Denoted by:
OE(bird nil)	Bird
OE(sparrow Bird)	Sparrow
OE(ernie Sparrow)	Ernie
OE(penguin Bird)	Penguin
OE(tweety Penguin)	Tweety

The AE was trained to encode the following assertions:

AE(Bird Fly)	AE((Bird Fly) Can)
AE(Sparrow Fly)	AE((Sparrow Fly) Can)
AE(Penguin Fly)	AE((Penguin Fly) Cannot)
AE(Ernie Fly)	AE((Ernie Fly) Can)
	AE((Ernie Fly) Cannot)
AE(Tweety Fly)	AE((Tweety Fly) Can)
	AE((Tweety Fly) Cannot)

For **Ernie** and **Tweety** both possible inferences (i.e., **Can** and **Cannot**) were generated for test purposes. The IN was trained to do the inferences:

```
AE(Bird Fly)     -> AE((Bird Fly) Can)
AE(Penguin Fly) -> AE((Penguin Fly) Cannot)
AE(Sparrow Fly) -> AE((Sparrow Fly) Can)
```

The relations encoded in the OE and the valid inferences in the IN can be visualized as (see Figure 3):

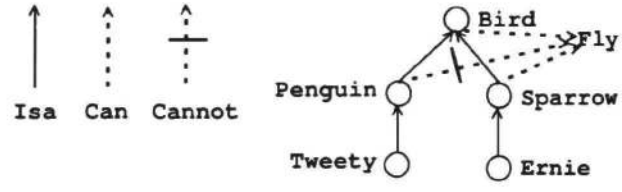

Figure 3: Graphical representation of the domain.

The main purpose of this simplified example is not to show that the architecture can handle both defaults and exceptions, but rather to relate the points made about context and content to a specific example. It, however, shows why the this particular network can generalize to the novel situations:

```
AE(Ernie Fly)    AE(Tweety Fly)
```

For visualization purposes the dimensionality of the hidden layer of the encoders was reduced to two units. The OE was a 12-2-12 sequential RAAM (i.e., the left input slot, in Figure 2, had a size of 10 units and the right had the same as the hidden layer). The representations for the atomic objects (i.e., **bird**, **sparrow**, etc.) were assigned a 10-element localistic non-overlapping representation. The AE was a 4-2-4 sequential RAAM, and the IN a 2-2 feed-forward network.

The hidden space for the encoded objects is shown in Figure 4(a). In this diagram, the hyperplanes for weights connected to the two output units (i.e., units 11-12, represented in the figure by 1 and 2 respectively) representing classes (i.e., **Bird**, **Sparrow** and **Penguin**) are

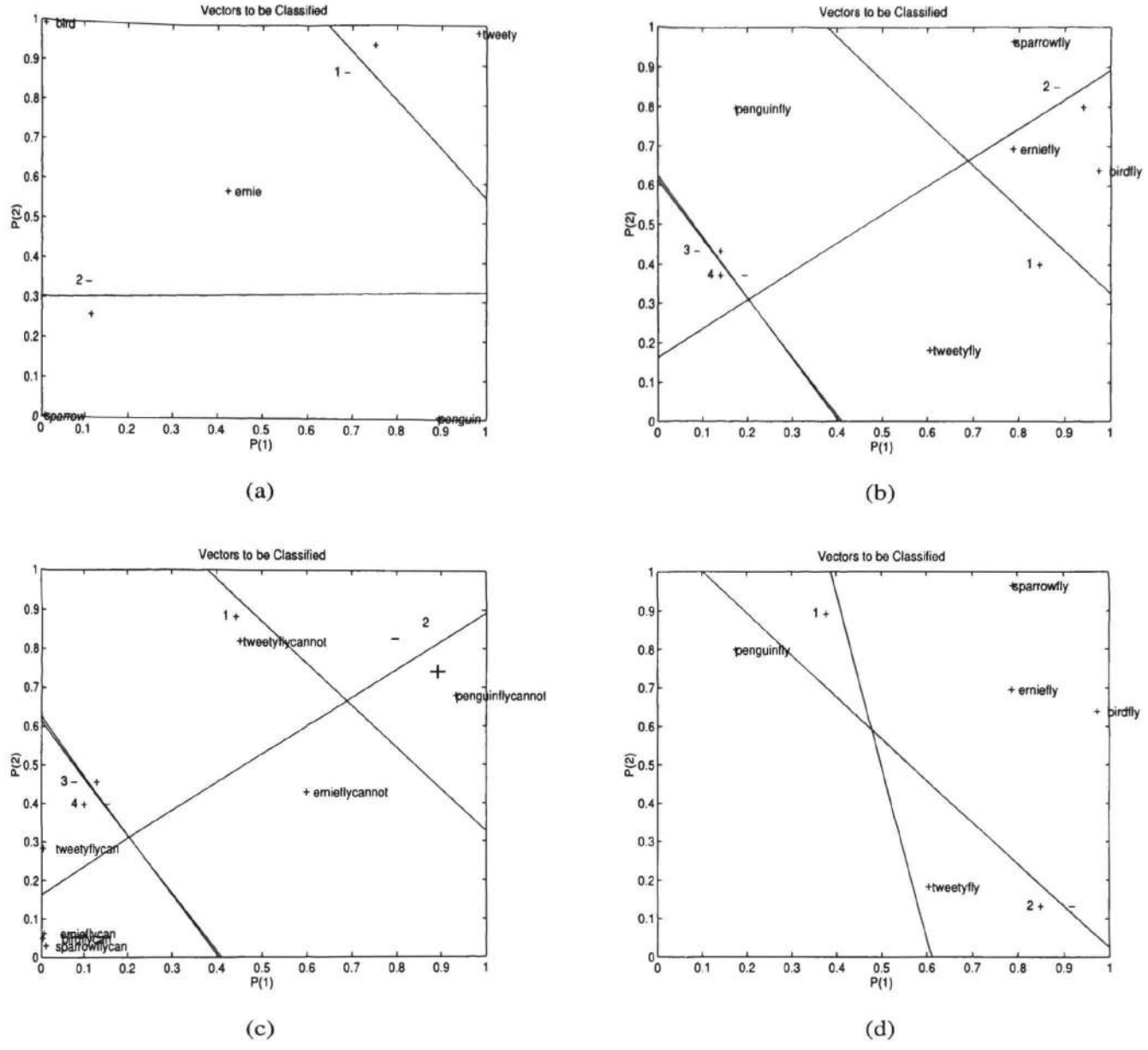

Figure 4: The hidden space of the OE (a), the AE (b and c) and the IN (d). Please note, that all the locations in the space (e.g., bird) actually are the generated representation after training (e.g., **Bird**).

also included. The first thing to note is that the representation for **Bird** (top left corner) has become close to [0 1]. This means that all members of the class **Bird** (i.e., **Penguin** and **Sparrow**) must end up on the negative side of the hyperplane enforced by the first class unit (i.e., unit 11), and the positive side of the other (i.e., unit 12). The representational region in the OE for members of the **Bird** class therefore becomes the region between the positive side of the second hyperplane and the x-axis. Similarly, the region for members of the **Sparrow** class (represented by [0 0]) becomes the region between the two hyperplanes, and the **Penguin** class the top-right region in the diagram. The reason that **Bird** ([0 1]) ends up in the **Sparrow** class region, is that the representation chosen for **nil** is [0 0] which is the same as the one developed for the **Sparrow** class. That, however, is not essential for the current purposes.

In the assertion space (Figure 4b and 4c) the representational regions for **Fly**, **Can** and **Cannot** are easy to identify. These regions are defined by output units 3 and 4 in the AE. **Fly** and **Cannot** are located to the positive side of 3 and negative side of 4, and **Can** vice versa. Moreover, the findings for the objects space are useful also in the assertion space, which in turn is the space used by the IN. It is possible to identify the region of the assertion space in which, for instance, new members of the **Penguin** class will end up. We can note that both units of the OE (the x and y axis in Figure 4a) will receive an activation above 0.5 for members of the **Penguin** class. This means that all new members of the class will be on the positive side of

the units 1 and 2 in the AE when combined with **Fly** (Figure 4b), i.e., the region in which **TweetyFly** now appears. Combining this with the hyperplanes of the IN (Figure 4d) it is possible to define the region for which the IN will make 'correct' inferences concerning **Penguin**. A location on the positive side of both hyperplanes means that the inference network will transform the location to one above [0.5 0.5] in the assertion space (Figure 4c), which always is classified as a no-**Fly** zone, by hyperplanes 3 and 4. One could also note that not all members of the **Penguin** class are guaranteed to actually end up the positive side of these hyperplanes. Some (e.g., those which receive an activation in the AE of about [1.0 0.3] which are likely to be transformed in the IN to a position close to 0 for the x axis and definitely below 0.5 for the y axis) novel member of the **Penguin** class will be classified as flying. We will in the following simulation see examples of this.

Context-dependent processing

Let us now turn to the remaining issue to be resolved; i.e., the impact of the context for solving practical problems. We will here refer to simulations reported elsewhere (cf. Bodén and Niklasson, 1999). In a series of simulations we examined the performance of the architecture on problems involving defaults and exceptions. Two contexts for some test objects (see D1 and D2 Figure 5) were used to evaluate the performance of the architecture. Of special interest was to evaluate the effect of the contextual feedback between the different sub-networks, allowing fully context-dependent representations.

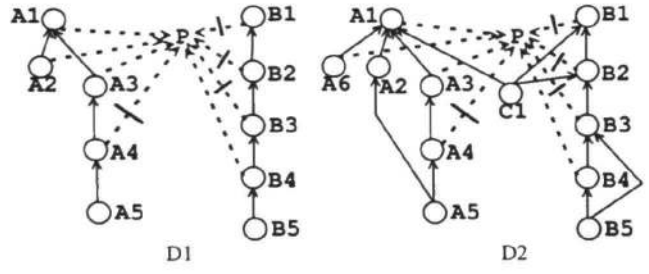

Figure 5: Two sample contexts, D1 and D2.

The architecture was trained on the two contexts (D1 and D2). After training, it was tested which content (**P+** or **P-**) was assigned to the test objects (**A5** and **B5** for both D1 and D2, and **C1** for D2). The content assigned by the IN was compared to the representations formed in the AE, and the one with the closest Euclidean distance was chosen. This can, for reasons explained earlier, be somewhat misleading but this approach do not favor one outcome over the other, which means that an average over several runs will give an objective result. In the first run on D1 the inference on **A5P** (i.e., the output of the IN) was 1.300 from **A5P+** and 0.116 from **A5P-**.

For D1, the size of the RAAMs used were OE 10-3-10, AE 6-3-6 and IN 3-3. For D2, the size were 12-3-12, 6-3-6 and 3-3. For each experiment 30 runs were conducted with contextual feedback enabled, and 30 with it disabled. Training was conducted for 10000 epochs with learning rate 0.1 and momentum 0.9. The results are listed in Table 1.

Table 1: Results from the D1 and D2 data sets

Context	Object	% P+	% P-	Contextual feedback
D1	A5	30	70	Yes
D1	A5	37	63	No
D1	B5	77	23	Yes
D1	B5	62	38	No
D2	A5	10	90	Yes
D2	A5	20	80	No
D2	B5	43	57	Yes
D2	B5	13	87	No
D2	C1	20	80	Yes
D2	C1	47	53	No

Some interesting observations can be made. Generally the architecture supports shortest path reasoning, e.g., for **A5** and **B5** in D1. Contextual feedback accentuates this preference, which is most obvious for **B5** in D2, where the path **B5->B3->P+** is as long as **B5->B4->P-**. The results show that architecture with feedback assigns positive or negative content with almost equal probability (without feedback 13% vs. 87%, and with feedback 43% vs. 57%). Compare this to **A5->A4->P-** and **A5->A2->A1->P+**, where the feedback has increased the bias for the shorter of the paths.

The most obvious reflection one can make, is that the effect on **C1** in D2 is rather dramatic. Without feedback the two outcomes occur with almost equal probability. With feedback the preference is for **P-**. This example can be compared to an extension of the famous Nixon diamond, i.e., Nixon is a quaker, republican and colonel. Quakers are pacifists, but republicans and colonels are not. One way of reasoning is that since the majority of categories of which Nixon is a member are non-pacifists, he is too.

Conclusion

We have argued that connectionism can offer alternatives to classical explanations for cognitive phenomena provided that content and context are defined in terms more natural to connectionist architectures, learning, weights and internal activations. Such definitions were provided and connected to an example.

If the approach we suggest is accepted, it is possible to explain not only context-independent reasoning (see Niklasson and van Gelder (1994) who used a related architecture for syntactic transformations), but also *context-dependent reasoning*, by referring the performance exhibited on data sets like D1 and D2.

Phillips (1998) noted that the networks used by Niklasson and van Gelder (1994) could not support systematicity at

the compositional level, only at the component level. The approach used in this paper shows that connectionist architectures can support systematicity at both levels, by incorporating contextual feedback.

We have shown how compositionality and context-dependence can co-exist within the same framework. The explanation we supply is based on weight regions expressing spatial structure which mirror contextual similarities among representations.

We also argued that connectionist and classicist systems should be allowed to make the same assumption about the example domain. Here two rather small data sets were used and cannot give the complete story but they can serve as useful indicators of what to look for. It would be quite easy to define an empirical investigation of how humans perform on D1 and D2. If the performances of humans differ significantly from the performance of our architecture, this would be quite damaging for our argument. If not, our view would be justified both on technical and empirical grounds.

Acknowledgment

This paper was made possible by a grant from The Foundation for Knowledge and Competence Development (1507/97), Sweden, to the first author, an Australian Research Council grant to the second author and a grant to both authors from the University of Skövde, Sweden.

References

Bodén, M. and Niklasson, L., (1999), Semantic systematicity and context in connectionist networks, (submitted to *Connection Science*, Carfax Publishers Ltd.).

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.

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

Goschke, T. and Koppelberg, D., (1991), The Concept of Representation and the Representation of Concepts in Connectionist Models, In Ramsey, W., Stich, S. and Rumelhart, D. E., (eds.), *Philosophy and Connectionist Theory*, LEA, pp. 129 - 162.

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., (1994a), Systematicity in Connectionist Language Learning, *Mind and Language*, vol. 9, no. 3.

Hadley, R. F., (1994b), Systematicity Revisited, Mind and Language, vol. 9, no. 4, Blackwell Publ., pp. 431 - 443.

Hadley, R. F. and Hayward, M. B., (1997) Strong Semantic Systematicity from Hebbian Connectionist Learning, *Mind and Machines*, 7, pp. 1 - 37.

Matthews, R. F., (1994), Three-Concept Monte: Explanation, Implementation, and Systematicity, *Synthese*, 101, Kluwer Academic Publishers, pp. 347 363.

Niklasson, L. F. and Sharkey N. E., (1992), Connectionism and the Issues of Compositionality and Systematicity, *Cybernetics and Systems Research*, Trappl (ed.), World Scientific, pp. 1367 1374.

Niklasson, L. F. and van Gelder, T., (1994), Can Connectionist Models Exhibit and Explain Non-Classical Structure Sensitivity, *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, LEA, pp. 664 - 669.

Palmer, S. E., (1987), Fundamental Aspects of Cognitive Representation, *Cognition and Categorization*, Rosch, E. and Lloyd, B. B., (eds.), LEA, Hillsdale, NJ, pp. 259 - 303.

Phillips, S., (1994), Strong Systematicity within Connectionism The Tensor Recurrent Network, *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, LEA, pp. 723 - 727.

Phillips, S., (1998), Are Feedforward and Recurrent Networks Systematic? Analysis and Implications for a Connectionist Cognitive Architecture, *Connection Science*, Carfax Publishing Ltd., vol 10, no 2, pp. 137 160.

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

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. and Niklasson L., (1994), Classicalism and Cognitive Architecture, *The Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, LEA, pp. 959 - 964.