Natural Language Recursion and Recurrent Neural Networks

Morten H. Christiansen
Philosophy-Neuroscience-Psychology Program
Department of Philosophy
Washington University in St. Louis
Campus Box 1073
One Brookings Drive
St. Louis, MO 63130
morten@twinearth.wustl.edu

Nick Chater
Neural Networks Research Group
Department of Psychology
University of Edinburgh
7 George Square
Edinburgh EH8 9JZ
Scotland, U.K.
nicholas@cogsci.ed.ac.uk

Abstract

The recursive structure of natural language was one of the principal, and most telling, sources of difficulty for associationist models of linguistic behaviour. It has, more recently, become a focus in the debate surrounding the generality of neural network models of language, which many would regard as the natural heirs of the associationist legacy. Can neural networks learn to handle recursive structures? If not, many would argue, neural networks can be ruled out *a priori* as viable models of language processing. In this paper, we shall reconsider the implications of natural language recursion for neural network models, and present a range of simulations in which recurrent neural networks are trained on very simple recursive structures. We suggest implications for theories of human language processing.

1 Introduction

From its inception, cognitive science has paradigmatically eschewed finite state models of natural language processing. The existence of complex recursive language structures—involving, most importantly, *center-embeddings* or *cross-dependencies*—appears to militate against finite state accounts of linguistic behaviour. Such sentence constructions are difficult to process because it is necessary to keep track of arbitrarily many different dependencies at once. For example, processing arbitrarily deep center-embedded constructions requires (at least) an unbounded ("last-in-first-out") stack; and processing an arbitrary number of cross-dependencies requires (at least) an unbounded ("first-in-first-out") queue. This is not possible for associationist accounts, which assume that the language processor is a (particular kind of) finite state machine (FSM). Similarly, assuming, as we must, that all parameters have finite precision, any finite neural network is also a finite state machine. Hence, it seems *prima facie* that neural network models of language processing also cannot account for the existence of recursive natural language constructions.

In this paper, we re-examine the problem of accounting for natural language recursion in neural networks *qua* finite state models of language processing. First, we discuss center-embedded and cross-dependency recursion as it occurs in natural language, arguing that while neural network models do not need to capture arbitrarily complex recursive structure, they must be able to handle recursive regularities of a limited depth. In section 3, we present a series of simulations involving recurrent neural

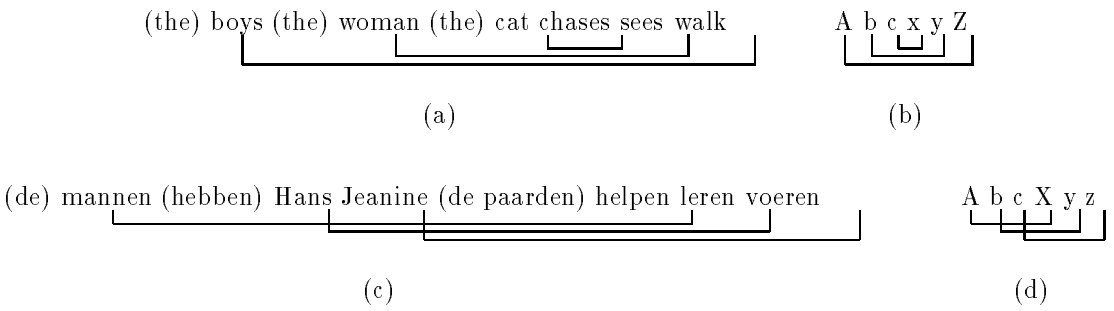


Figure 1: Illustration of a center-embedded sentence in English (a), a Dutch sentence with cross-dependencies (c), and the abstract structure of their respective dependency relations ((b) and (d)).

networks trained on center-embedded and cross-dependency structures. Finally, we compare network performance with experimentally observed limitations on human processing of similar structures, and suggest implications for theories of human language processing.

2 Recursion in Natural Language

Center-embedding and cross-dependency recursion have played the leading roles in the dismissal of finite state models of language (e.g., in Chomsky, 1957). Examples of such constructions are shown in Figure 1(a) and (c). Center-embedding occurs in a large number of languages, including English; cross-dependencies are much less common, although clear examples occur in Swiss German and, as here, in Dutch. As is clear from 1(a), even a sentence with two center-embeddings is very difficult to process. 1(a) can be loosely glossed "the boys, seen by the woman who is chased by the cat, walk". The literal English translation of 1(c) is 'the men (have) Hans Jeanine (the horses) help teach feed' and can be glossed as 'the men helped Hans teach Jeanine to feed the horses'. The syntactic manifestation of the dependency of interest in these examples is the singular/plural agreement between the subject-nouns and their corresponding verbs. Figure 1(b) and (d) show the abstract structure of this agreement, with the convention that upper case letters denote "plural" items, and lower case letters denote "singular" items.

One line of defense against the demise of finite state models of language processing is that arbitrarily long center-embedded constructions, while allowed by the rules of generative grammar, do not occur in practice (Christiansen, 1992). Indeed, empirical studies (e.g., Bach, Brown & Marslen-Wilson, 1986) have shown that sentences with either three or more center-embeddings or three or more cross-dependencies are universally hard to process and understand. Perhaps, then, FSMs, including neural networks, might be able to model language processing successfully. A second line of defense is that all real computational devices, including the digital computers on which successful symbolic parsers for recursive structures are routinely constructed are, at bottom, finite state devices. Moreover, the brain itself has a finite number of states (again assuming that we do not advert to arbitrary precision), so the limitation which applies to neural networks must in any case apply to any cognitive model.

There is, however, a more sophisticated form of the original argument, based on the observation that what is important about generative grammar is not that it allows arbitrarily long and complex strings, but that it gives a simple set of rules which capture regularities in natural language. An adequate model of language processing must somehow embody such grammatical knowledge. It must, for example, be able to handle novel sentences which conform to the linguistic regularities, and be able to reject as ungrammatical novel strings which do not. In traditional computational linguistics, this is done by representing grammatical information and processing operations in terms of symbolic rules. While these rules could, in principle, be applied to sentences of arbitrary length and complexity, in practice they are necessarily bounded by the finiteness of the underlying hardware. Unless neural networks can perform the same trick of capturing the underlying recursive structures in language, then they cannot be complete models of natural language processing.

Importantly, this more sophisticated argument removes the debate concerning neural networks and natural language recursion from the domain of *a priori* speculation. It poses a specific challenge: to show that neural networks can capture the recursive regularities of natural language, while granting that arbitrarily complex sentences cannot be handled. We shall make a step towards addressing this challenge in the simulations below.

3 The Processing of Recursion in Recurrent Networks

The issue of recursion has been addressed before within a connectionist framework. For example, both Elman (1991) and Cleeremans, Servan-Schreiber & McClelland (1991) have demonstrated the ability of Simple Recurrent Networks (SRN) to deal with right recursive structures (which can, however, be handled by an FSM) as well as limited instances of center-embedded recursion. In addition, the latter form of recursion has been studied further by Weckerly & Elman (1992). SRN studies deliberately sidestep the goal of teaching the network to develop explicit representations of linguistic structure. Instead the network is trained to predict the next item in the sequence, and must learn the grammatical structure to do this. It is fair to say that these models have so far reached only a modest level of performance. Only little headway has been made towards more complex grammars involving center-embedded recursion (most noticeably by Elman, 1991 and Weckerly & Elman, 1992), but not towards cross-dependency recursion. The simulations reported in this paper build on and extend this work, by focussing directly on center-embedded and cross-dependency constructions, in the form that they were originally outlined by Chomsky (1957).[1]

3.1 Method

Recurrent networks are usually trained by "unfolding" them into feedforward networks with the same behaviour. The hidden units from the previous time-step are then treated as an additional set of inputs, allowing the resulting feedforward network to be trained using standard back-propagation. There are a various ways in which this unfolding can be achieved (see Chater & Conkey, 1992). One approach is

[1] In Christiansen & Chater (in prep), we also consider another simpler, recursive construction (which we call "counting recursion") introduced by Chomsky (1957) as falling outside the scope of finite state models.

to unfold the network through several time steps (Rumelhart, Hinton & Williams, 1986) so that each weight has several "virtual incarnations" and then back-propagate error through the resulting network. The overall weight change is simply the sum of the changes recommended for each incarnation. This "back-propagation through time"—or, Recurrent Back-Propagation (RBP)—is typically implemented by unfolding through a small number of time steps (7 for the current simulations). The copy-back scheme employed in SRNs can be viewed as a special case of RBP, in which the back-propagation of error stops at the first copy of the hidden units—the context units. Simulations by Chater & Conkey (1992) have shown that RBP performs better than SRNs on a number of tasks (such as, learning to be a delay line and performing discrete XOR), although the former is considerably more computationally expensive. A secondary motivation for the present simulations is therefore to compare the two training regimes on more language-like tasks[2].

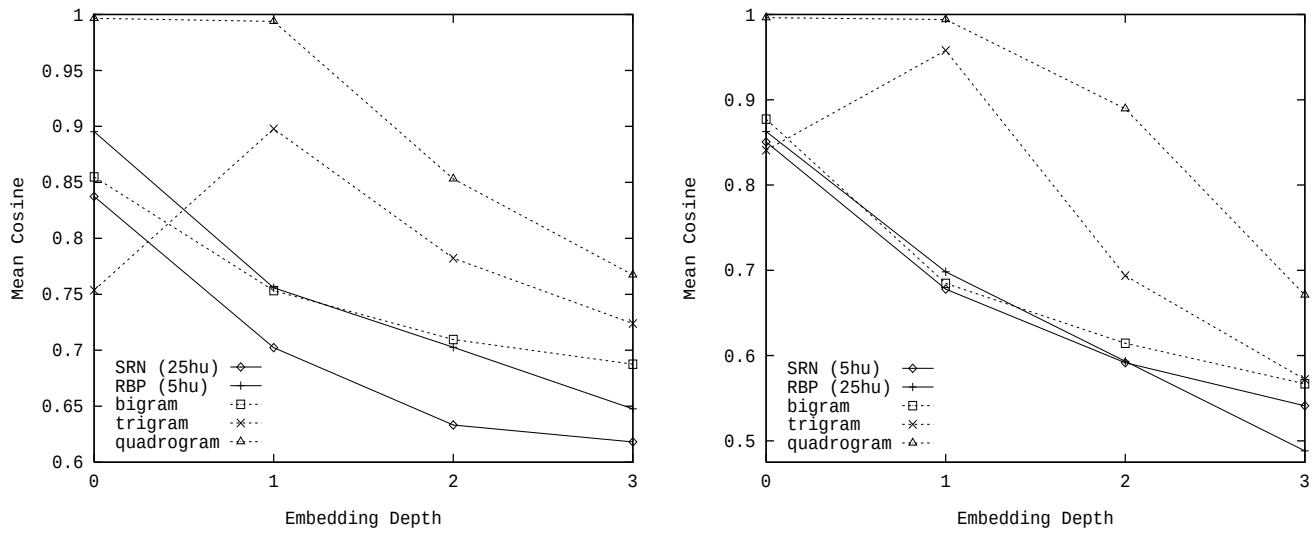
As a benchmark on which to assess the performance of the two networks, we also developed a simple statistical prediction method, based on n-grams, strings of n consecutive words. The program is "trained" on the same stimuli used by the networks, and simply records the frequency of each n-gram in a look-up table. It makes predictions for new material by considering the relative frequencies of the n-grams which are consistent with the previous $n-1$ words. The prediction is a vector of relative frequencies for each possible successor item, scaled to sum to 1, so that they can be interpreted as probabilities, and are therefore directly comparable with the output vectors produced by the networks. Below, we report the predictions of bigram, trigram and quadrogram models and compare them with the network models.

To construct the stimuli on which to test the networks, we used the structures shown in Figure 1(b) and (d). That is, the first half of a string consists of "noun" type words (*A, a, ...*); the second half of the string consists of "verb" type words (*...z, Z*). Each word has singular and plural forms, lower case being used for singular items and upper case for plural. Furthermore, each sentence has an end of sentence marker, a "." , after the final "verb". Thus, the set of center-embeddings strings include: *AZ., aazz., aAZz., AazZ., AaazzZ., aaAZzz., ...*; cross-dependency strings include *AZ., aazz., aAzZ., AaZz., AaaZzz., aaAzzZ., ...* For both experiments training sets of 2000 sentences and test sets of 1000 sentences were generated in a probabilistic fashion (each with a mean sentence length of about 4.7 and sd about ± 1.3)[3]. The next two subsections report the results obtained in two experiments using these two languages involving, respectively, a two word and an eight word vocabulary[4].

[2] In any interesting language-like task, the next item will not be deterministically specified by the previous items, and hence it is appropriate for the prediction to take the form of a probability distribution of possible next items. Consequently, network performance in the simulations reported below was measured against this probability distribution directly, rather than against predictions of the specific next item in the sequence. Following Elman (1991; and others) the mean cosine between network output vectors and probability vectors given previous context is used as a quantitative measure of performance.

[3] For each sentence the depth of nesting was computed by iterating the following: if $r < p^n(1-p)$ then an extra level of nesting would be added to the sentence, where r is a random number between 0 and 1; p the probability of adding a level of nesting (0.3 in the simulations reported here); and n the number of nestings that the sentence already has.

[4] Initial explorations indicated that the best performance for the SRNs was to be obtained with a learning rate of 0.5, a momentum of 0.25 and an initial randomization of the weights between ± 0.5. In the case of RBP, no momentum was used, the learning rate was set to 0.5 and the weights initialized randomly between ± 1.0.



(a) (b)

Figure 2: Network and n-gram performance on (a) center-embedded recursion and (b) cross-dependency recursion plotted as a function of embedding depth (2 word vocabulary).

3.2 Experiment 1: Two Word Vocabulary

The first experiment involves the two word vocabulary as found in Chomsky (1957) (with the additional singular/plural agreement constraint as described in section 3.1). Thus, we have a vocabulary consisting of a "noun" in a singular and plural form, *'a'* and *'A'* respectively, and a "verb" likewise in a singular and plural form, *'z'* and *'Z'* respectively. Both networks were trained with 5, 10 and 25 hidden units on each of the two tasks, but only the results of the network configurations with the best performance is reported here[5]. The inputs and outputs were represented as binary localist vectors with one bit for each word form and one for the end of sentence marker (totalling 5 inputs/outputs)[6].

Center-embedded recursion: In the first task, the best RBP network (with 5 hidden units) performed slightly better on this task than the best SRN (with 25 hidden units)[7]. This is also mirrored in the relative increase in performance through learning with an 47% improvement of performance for the RBP net (from $cos0.538$ before training to $cos0.792$) compared with a 34% improvement for the SRN (from $cos0.560$ to $cos0.755$ after training). Turning to Figure 2(a), we can see that the RBP net also had a slightly better performance than the SRN across embedding depth. Moreover, the performance degraded over embedding depth for both networks. Note that the performance of the two nets are comparable with the performance based on bigram predictions, but inferior to that of both trigrams

[5]The differences in performance between the three network configurations was small for both nets. For a full report, see Christiansen & Chater (in prep).

[6]Through cross-validation it was found that the number of epochs necessary to reach peak performance in both nets varied with the size of the hidden unit layer. Increasing the hidden unit layer resulted in faster training (although the RBP nets exhibited much faster training across the board). Subsequently, the SRNs with 5, 10 and 25 hidden units were trained for 500, 450 and 350 epochs, respectively. The RBP network with 5, 10 and 25 hidden units were trained for 275, 250 and 200 epochs, respectively.

[7]The level of performance displayed by both nets was below what Elman (1991) has reported (mean $cos0.852$), but well above the performance obtained by Weckerly & Elman (1992) on center-embedded recursive structures.

and quadrograms.

Cross-dependency recursion: In the second task, both networks did equally well on this task and obtained the same relative increase in performance through learning (SRN with 5 hidden units: 29% – from *cos*0.571 to *cos*0.737; RBP with 25 hidden units: 29% – from *cos*0.570 to *cos*0.741). This trend continues in Figure 2(b), which also illustrates the close relationship in which both network and bigram predictions follow the same degradation pattern across embedding depth (as in the previous task). Furthermore, the trigram and quadrogram based predictions are again superior to network predictions.

3.3 Experiment 2: Eight Word Vocabulary

In order to test the ability of both networks to capture the recursive regularities necessary for dealing with novel sentences, we conducted a second experiment involving an eight word vocabulary[8]. Thus, we have four "nouns" in a singular (*'a', 'b', 'c', 'd'*) and a plural form (*'A', 'B', 'C', 'D'*), an four "verbs" likewise in a singular (*'w', 'x', 'y', 'z'*) and a plural form (*'W', 'X', 'Y', 'Z'*). In experiment 1, we found that the size of the hidden unit layer did not appear to influence performance on either of the tasks. We therefore decided only to train networks with 20 hidden units in the present experiment. Pilot studies indicated that the localist representation of words that we used in the previous experiment was inappropriate for the present experiment. We therefore adopted a different representation scheme in which each word was represented by a single bit (independently of its form) and the form was represented by one of two bits (common to all words) signifying whether a word was singular or plural. Thus, for each occurrence of a word two bits would be on—one bit signifying the word and one bit indicating its number[9]. The input/output consisted of 11 bit vectors (one for each of the eight words, one for each of the two word forms, and one for the end of sentence marker). To allow assessment of network performance on novel sentences, we introduced two extra test sets with, respectively, 10 novel sentences and 10 previously seen sentences (mean: 5.3; sd: ± 1.6).

Center-embedded recursion: On this task the SRN performed modestly better than the RBP network—though the latter had a much better relative performance improvement through learning (RBP: 67% – from *cos*0.464 to *cos*0.776; SRN: 35% – from *cos*0.602 to *cos*0.813). Figure 3(a) shows that performance as a function of embedding depth exhibits much the same general pattern of degradation as found on the same task in the previous experiment (except from a minor peak on depth 1). Once again, we see that the performance of the nets is comparable with that of bigram predictions, but this time the trigram and quadrogram based performance is less predominant over network performance than in the experiment 1. Most importantly, the networks showed no significant difference

[8] This extension of the vocabulary was necessary, since leaving out certain sentence structures in the previous experiment would have skewed the training set in a problematic fashion. Moreover, we wanted to investigate how the networks would perform on a bigger vocabulary.

[9] It is worth noticing that this kind of representational format appears more plausible than a strict localist one. In particular, it is unlikely that we 'store' singular and plural forms of the same word (e.g., "cat" and "cats") as distinct and completely unrelated representations as it would be the case with localist representations. Rather, we would expect the human language processing mechanism to take advantage of the similarities between the two word forms to facilitate processing.

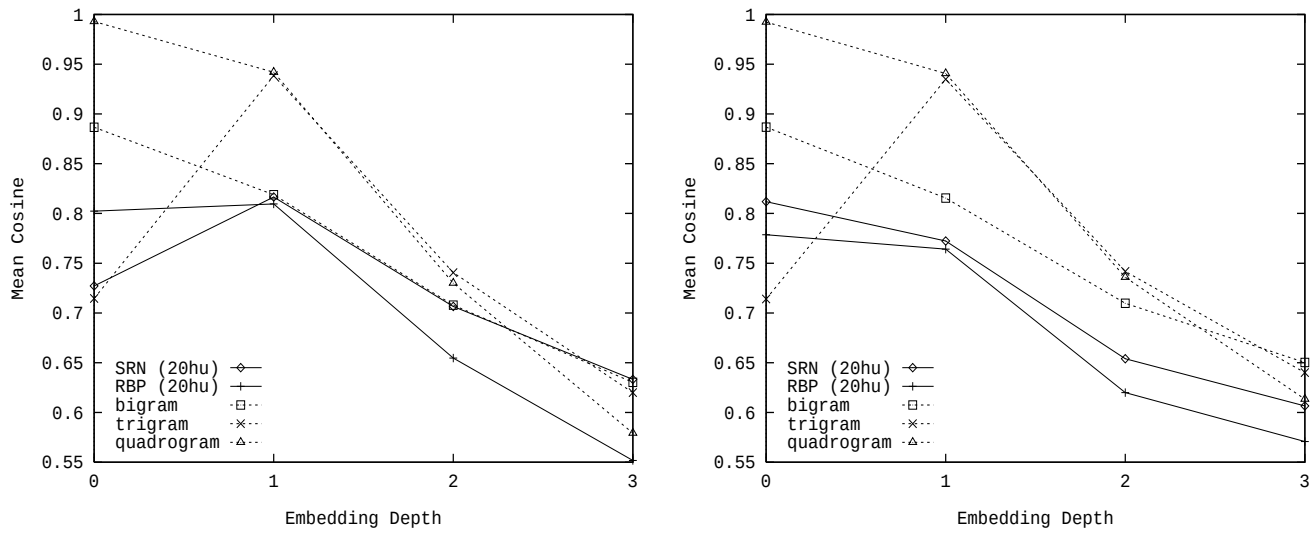


Figure 3: Network and n-gram performance on (a) center-embedded recursion and (b) cross-dependency recursion plotted as a function of embedding depth (8 word vocabulary).

in performance on, respectively, the novel and the previously seen test sentences (in fact, the RBP network performed slightly better on the novel sentences compared with previously seen sentences, suggesting that the network might have been somewhat undertrained). Thus, the SRN obtained a $cos0.555$ on the novel sentences and a $cos0.550$ on the sentences it already has been exposed during training[10]. The RBP network reached a $cos0.437$ on the novel sentences compared with a $cos0.399$ on the previously seen sentences.

Cross-dependency recursion: The overall performance of the two nets on the final task was much alike, though favouring the SRN. This is in contrast to the relative increase in performance achieved through learning, where the RBP network obtained a 58% improvement (from $cos0.476$ to $cos0.755$) compared with the SRN's 28% (from $cos0.604$ to $cos0.773$). Figure 3(b) illustrates the close fit between the performance of the two networks across embedding depth. It also shows that the nets are not as close to the bigram performance as in the previous task (and in the previous experiment). Moreover, net performance is still inferior to trigram and quadrogram based performance. Yet, as it was the the case in the previous task, both networks were able to deal with novel sentences, indicating that they had learned the underlying recursive regularities. The SRN reached an overall performance of $cos0.538$ on the novel sentences and $cos0.549$ on the sentences it had already seen. For the RBP network, $cos0.476$ was accomplished on the novel test sentences and $cos0.463$ on the previously seen sentences.

[10]Note that this apparently low performance is due to the fact that it was measured against the probability distribution of these two sets, whereas the nets had been trained on (and, thus, become sensitive to) the much more complex probability distribution of the 2000 sentences in the training set.

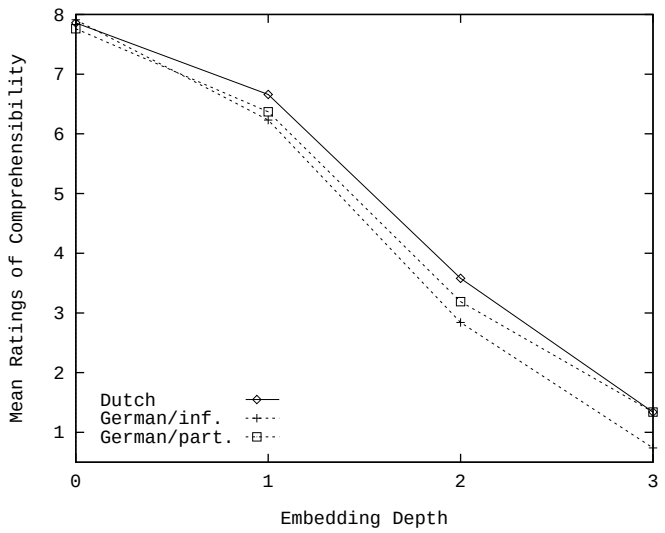


Figure 4: The performance of native speakers of German and Dutch on, respectively, center-embedded sentences and sentences involving cross-dependencies is plotted against embedding depth. The figure is based on data reported in Bach, Brown, & Marslen-Wilson (1986).

4 Discussion

In this paper we posed the following challenge: Can neural networks capture the recursive regularities of natural language if we accept that arbitrarily complex sentences cannot (and, perhaps, should not) be handled? The ability of both kinds of networks to generalise to novel sentences involving either center-embedded or cross-dependency recursion in experiment 2 suggests that neural networks might be able to do the trick. But where does that leave the pattern of gradual breakdown of performance as observed in all the simulations presented here? If we compare this breakdown pattern with the degradation of human performance on center-embedded and cross-dependency structures (as can be adduced from Figure 4[11]), we can conclude that such a breakdown pattern is, indeed, desirable from a psycholinguistic perspective. Thus, network (and bigram based) performance across embedding depth appears to mirror general human limitations on the processing of complex recursive structures.

Two other things are worth noticing. First of all, the overall performance (of both nets and n-gram based predictions) on the cross-dependency recursion task was somewhat better than expected. This is a positive result, given that dealing with cross-dependency structures requires the acquisition of (something closely related to) a context-sensitive grammar, whereas center-embedded recursion 'merely' requires the acquisition of a context-free grammar. The networks, then, did better on the cross-dependency task than was to be expected given the structural complexity of the learning task. This is important, since human performance seems to be quite similar on both kind of recursive structure (see Figure 4). Secondly, there was no significant performance difference between the two kind of networks on either of the tasks (in both experiments). This means that the negative results reported by Chater & Conkey (1992) regarding SRN performance on certain non-language tasks do

[11] The data from Bach, Brown & Marslen-Wilson (1986: p. 255, table 1: test results) is displayed using $f(x) = 9 - x$ to facilitate comparisons with net and n-gram performance expressed in terms of mean cosines.

not extend themselves to more language-like tasks. Thus, in addressing our secondary motivation for the present simulations, we found, rather surprisingly, that unfolding a recurrent network for the purpose of RBP does not seem to provide additional computational power on language-like tasks such as center-embedded and cross-dependency recursion.

The close similarity between the breakdown patterns in human and neural network performance on complex recursive structures supports two wide-reaching conjectures. On the one hand, neural network models—in spite of their finite state nature—must be considered as viable models of natural language processing. At least, we have shown that the existence of center-embedding and cross-dependency no longer can be used as *a priori* evidence against neural network (and other finite state) models of linguistic behaviour. On the other hand, the common pattern of graceful degradation also suggests that humans, like neural networks, are sensitive to the statistical structure of language. Neural networks pick up certain simple statistic contingencies in the input they receive (the simulations presented here indicate that such statistics might resemble bigram based probability distributions). We suggest that the breakdown pattern in human performance on complex recursive structures also might be due to a strong dependence on such statistics in the acquisition of linguistic structure. Whether these conjectures are true is a matter of future empirical research, not *a priori* speculation.

References

Bach, E., Brown, C. & Marslen-Wilson, W. (1986) Crossed and nested dependencies in German and Dutch: A psycholinguistic study. *Language and Cognitive Processes*, **1**, 249–262.

Chater, N. & Conkey, P. (1992) Finding Linguistic Structure with Recurrent Neural Networks. In *Proceedings of the Fourteenth Annual Meeting of the Cognitive Science Society*, Indiana University, Bloomington, July/August.

Chomsky, N. (1957) *Syntactic Structures*. The Haque: Mouton.

Christiansen, M. (1992) The (Non)Necessity of Recursion in Natural Language Processing. In *Proceedings of the Fourteenth Annual Meeting of the Cognitive Science Society*, Indiana University, Bloomington, July/August.

Christiansen, M. & Chater, N. (in preparation) Finite State Models of Language Learning: A Connectionist Perspective. Ms.

Elman, J. L. (1991) Distributed Representation, Simple Recurrent Networks, and Grammatical Structure. *Machine Learning*, **7**, 195–225.

Rumelhart, D., McClelland, J. & Williams, R. (1986) Learning Representations by back-propagating errors. *Nature*, **323**, 533–536.

Servan-Schreiber, D., Cleeremans, A. & McClelland, J. L. (1991) Graded state machines: The representation of temporal contingencies in simple recurrent networks. *Machine Learning*, **7**, 161–193.

Weckerly, J. & Elman, J. (1992) A PDP Approach to Processing Center-Embedded Sentences. In *Proceedings of the Fourteenth Annual Meeting of the Cognitive Science Society*, Indiana University, Bloomington, July/August.