Simple Recurrent Networks and Natural Language: How Important is Starting Small?

Douglas L. T. Rohde[1,3] **David C. Plaut**[1,2,3]
[1]School of Computer Science and [2]Department of Psychology, Carnegie Mellon University,
and the [3]Center for the Neural Basis of Cognition

Abstract

Prediction is believed to be an important component of cognition, particularly in the processing of natural language. It has long been accepted that recurrent neural networks are best able to learn prediction tasks when trained on simple examples before incrementally proceeding to more complex sentences. Furthermore, the counter-intuitive suggestion has been made that networks and, by implication, humans may be aided during learning by limited cognitive resources (Elman, 1991 *Cognition*). The current work reports evidence that starting with simplified inputs is not necessary in training recurrent networks to learn pseudo-natural languages. In addition, delayed introduction of complex examples is often an impediment to learning. We suggest that special teaching methods and limited cognitive resources during development may be of no benefit for learning the structure of natural language.

Introduction

The question of how humans are able to learn a natural language despite the apparent lack of adequate feedback has long been a perplexing one. Baker (1979) argued that children do not receive a sufficient amount of negative evidence to properly infer the grammatical structure of language. Computational theory suggests that this is indeed problematic, as Gold (1967) has shown that, without negative examples, no superfinite class of languages is learnable. The class of regular languages is superfinite, as are context-free and context-sensitive languages. Therefore, unless the set of possible natural languages is highly restricted, it would appear that such languages are not learnable from positive examples. How, then, are humans able to learn language? Must we rely on extensive innate knowledge?

In fact, a frequently overlooked source of information is the statistical structure of natural language. Language production can be viewed as a stochastic process—some sentences and grammatical constructions are more likely than others. The learner can use these statistical properties as a form of implicit negative evidence. Indeed, *stochastic* regular languages and stochastic context-free languages *are* learnable using only positive data (Angluin, 1988). One way the learner can take advantage of these statistics is by attempting to predict the next word in an observed sentence. By comparing these predictions to the actually occurring next word, feedback is immediate and negative evidence derives from consistently incorrect predictions. Indeed, a number of studies have found empirical evidence that humans do generate expectations in processing natural language and that these play an active role in comprehension (Neisser, 1967; Kutas & Hillyard, 1980; McClelland & O'Regan, 1981).

Elman (1991, 1993) provided an explicit formulation of how a learning system might infer the grammatical structure of a language on the basis of performing a word prediction task. He trained a simple recurrent network to predict the next word in sentences generated by an English-like artificial grammar with number agreement, variable verb argument structure, and embedded clauses. He found that the network was able to learn the task but only if the training regimen or the network itself was in some way restricted in its complexity initially (i.e., it "started small"). Specifically, the network could learn the task either when it was trained first on simple sentences (without embeddings) and only later on a gradually increasing proportion of complex sentences, or when it was trained on sentences drawn from the full complexity of the language but with an initially faulty memory for context which gradually improved over the course of training. By contrast, when the network was given fully accurate memory and trained on the full complex grammar from the outset, it failed to learn the task. Elman suggested that the limited cognitive resources of the child may, paradoxically, be necessary for effective language acquisition, in accordance with Newport's (1990) "less is more" proposal.

This paper reports on attempts to replicate of some of Elman's findings using similar networks but more sophisticated languages. In contrast with his results, it was found that networks were able to learn quite readily even when confronted with the full complexity of language from the start. Under no circumstances did starting with simple sentences reliably aid learning and, in most conditions, it proved to be a hindrance. Furthermore, starting with the full language was of greater benefit when the grammar was made more English-like by including statistical constraints between main clauses and embeddings based on lexical semantics. We argue that, in the performance of realistic tasks including word prediction in natural language, recurrent networks inherently extract simple regularities before progressing to more complex structures, and no external manipulation of the training regimen or internal memory is required to induce this property. Thus, the work calls into question support for the claim that initially

S	→	NP VI . | NP VT NP .
NP	→	N | N RC
RC	→	who VI | who VT NP | who NP VT
N	→	boy | girl | cat | dog | Mary | John | boys | girls | cats | dogs
VI	→	walks | bites | eats | barks | sings | walk | bite | eat | bark | sing
VT	→	chases | feeds | walks | eats | bites | chase | feed | walk | eat | bite

Table 1: The underlying context-free grammar. Transition probabilities are specified and additional constraints are applied on top of this framework.

Verb	Intransitive Subjects	Transitive Subjects	Objects if Transitive
chase	-	any	any
feed	-	human	animal
bite	animal	animal	any
walk	any	human	dog
eat	any	animal	human
bark	only dog	-	-
sing	human or cat	-	-

Table 2: Semantic constraints on verb usage. Columns indicate legal subject nouns when verbs are used transitively or intransitively and legal object nouns when transitive.

limited cognitive resources or other maturational constraints are required for effective language acquisition.

Simulation Methods

We begin by describing the grammars used in both Elman's work and the current study. We then describe the corpora generated from these grammars, the architecture of the simple recurrent networks trained on the corpora, and the methods used in their training.

Grammars

The languages used in this work are similar in basic structure to that used by Elman (1991), consisting of simple sentences with the possibility of relative-clause modification of nouns. Elman's grammar involved 10 nouns and 12 verbs, plus the relative pronoun who and an end-of-sentence marker. Four of the nouns were transitive, four intransitive, and four optionally transitive. Six of the nouns and six of the verbs were singular, the others plural. Number agreement was enforced between nouns and verbs where appropriate. Finally, two of the nouns were proper and could not be modified.

This language is of interest because it forces a prediction network to form representations of potentially complex syntactic structures and to remember information, such as whether the noun was singular or plural, across long embeddings. Elman's grammar, however, was essentially purely syntactic, involving no form of semantics. Thus, the singular verbs all acted in the same way; likewise for the sets of plural verbs and singular and plural nouns. Natural language is clearly far more complex and the addition of semantic relationships ought to have a profound effect on the manner in which a language is processed.

The underlying framework of the grammar used in this study, shown in Table 1, is nearly identical to that designed by Elman. They differ only in that the current grammar adds one pair of mixed transitivity verbs and that it allows relative clauses to modify proper nouns. However, several additional constraints are applied on top of this framework. Primary among these, aside from number agreement, is that individual nouns can engage only in certain actions and that transitive verbs can operate only on certain objects. For example, anyone can walk intransitively, but only humans can walk something else and the thing walked must be a dog. These constraints are listed in Table 2.

Another restriction is that proper nouns cannot act on themselves. For example Mary chases Mary would not be a legal sentence. Finally, constructions of the form Boys who walk walk are disallowed because of semantic redundancy. These and the above constraints always apply within the main clause of the sentence. Aside from number agreement, which affects all nouns and verbs, the degree to which the constraints apply between a clause and its subclause is variable. In this way the correlation between a noun and its modifying phrase, or the level of information (about the identity of the noun) in the phrase, can be manipulated.

The basic structure shown in Table 1 becomes a stochastic context-free grammar (SCFG) when probabilities are specified for the various productions. Additional structures were also added to allow direct control of the percentage of complex sentences generated by the grammar and the average number of embeddings in a sentence. Finally, a program was developed which takes the grammar, along with the additional syntactic and semantic constraints, and generates a new SCFG with the constraints incorporated into the context-free transitions. In this way, a single SCFG can be generated for each version of the grammar. This is convenient not only for generating example sentences but also because it allows us to determine the optimal prediction behavior on the language. Given the SCFG and the sentence context up to the current point, it is possible to produce the theoretically optimal prediction of the next word. This prediction is in the form of a probability distribution over the 26 words in the vocabulary. The ability to generate this prediction, and hence to model the grammar, is what we expect the networks to learn.

Corpora

In order to study the effect of varying levels of information in embedded clauses, five classes of grammar were constructed. In class A, semantic constraints do not apply between a clause and its subclause, only within a clause. In class B, 25% of the

subclauses respect the semantic constraints, in class C, 50%, in class D, 75%, and in class E all of the subclauses are constrained. Therefore, in class A, which is most like Elman's grammar, the contents of a relative clause provide no information about the noun being modified other than whether it is singular or plural, whereas class E produces sentences which are the most English-like.

Elman (1991) first trained his network on a corpus of 10,000 sentences, 75% of which were complex. He reported that the network was "unable to learn the task" despite various choices of initial conditions and learning parameters. Three additional corpora containing 0%, 25%, and 50% complex sentences were then constructed. When trained for 5 epochs on each of the corpora in increasing order of complexity, the network "achieved a high level of performance." As in Elman's experiment, four versions of each class were created in the current work in order to produce languages of increasing complexity. Grammars A_0, A_{25}, A_{50}, and A_{75}, for example, produce 0%, 25%, 50%, and 75% complex sentences, respectively. In addition, for each level of complexity, the probability of relative clause modification was adjusted to match the average sentence length in Elman's corpora.

For each of the 20 grammars (five classes of semantic constraints by four percentages of complex sentences), two corpora of 10,000 sentences were generated, one for training and the other for testing. Corpora of this size are quite representative of the statistics of the full language for all but the longest sentences, which are relatively infrequent. Sentences longer than 16 words were discarded in generating the corpora, but these were so rare ($< 0.2\%$) that their loss should have negligible effects. In order to perform well, a network could not possibly "memorize" the training corpus but must learn the structure of the language.

Network Architecture

The architecture of the simple recurrent network used both by Elman and in the current work is illustrated in Figure 1. The network contained 6,936 trainable weights and included a fully connected projection from "context" units whose activations are copied from hidden units at the previous time step. The 26 inputs were encoded using basis vectors. One word was presented on each time step. Although the desired output of the network is a probability distribution indicating the expected next word, the target output during training consisted of the actual next word occurring in the sentence.

The current simulations were performed with *softmax* constraints (Luce, 1986) which normalize the output vector to a sum of 1.0, as opposed to the sigmoidal output units used by Elman. The divergence error measure (Hinton, 1989) was used in providing feedback to the network. The error for each unit is given by $d(\log d - \log y)$, where d is the target value and y is the output unit activation. Note that when the target is 0, this value is by convention 0 as well. Therefore, error is only injected at the unit representing the actual next word in the sentence, which is perhaps more plausible than other functions which provide feedback on every word in the vocabulary. Errors were not backpropagated through time, only through the current time step, and were therefore also relatively local in time. Hidden layer activation was not reset between sentences; however, the end-of-sentence marker clearly denotes a sentence boundary.

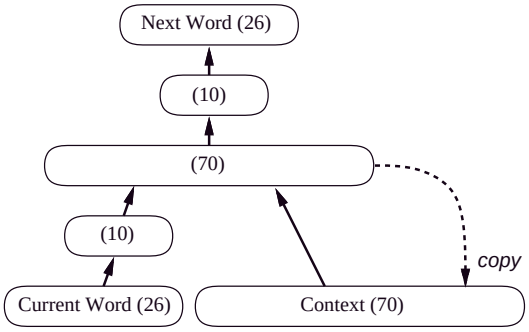

Figure 1: Network architecture. Each solid arrow represents full connectivity between layers (with numbers of units in parentheses). Hidden unit states are copied to corresponding context units (dashed arrow) after each word is processed.

Experiments

For each of the five language classes, two training regimens were carried out. In the *complex* regimen, the network was trained on the most complex corpus (75% complex) for 25 epochs with a fixed learning rate. The learning rate was then reduced to 0.0003 and the network was trained for one final pass through the corpus. In the *simple* regimen, the network was trained for five epochs on each of the first three corpora in increasing order of complexity. It was then trained on the fourth corpus for 10 epochs, followed by a final epoch at the reduced learning rate. The final six epochs of training on the fourth corpus (not included in Elman's design) were intended to allow performance with the simple regimen to reach asymptote. The network was evaluated on the test corpus produced by the same grammar as the final training corpus.

A wide range of training parameters were searched before finding a set which consistently achieved the best performance under nearly all conditions. The network used momentum descent with a momentum of 0.9, a learning rate of 0.004, and initial weights sampled uniformly between ± 1.0. Softmax output constraints were applied with a divergence error function. By contrast, the parameters selected by Elman included no momentum, a learning rate of 0.1 annealed to 0.06, and initial weights in the ± 0.001 range; also, softmax constraints were not used and sum-squared error was employed during training.

Both complex and simple trials were run for each of the five grammar classes. Twenty replications of each condition were performed, resulting in 200 total trials. Although the actual next word occurring in the sentence served as the target output during training, the network was expected to produce a distribution over all possible words. The target vectors in the testing corpora consisted of the theoretically correct pre-

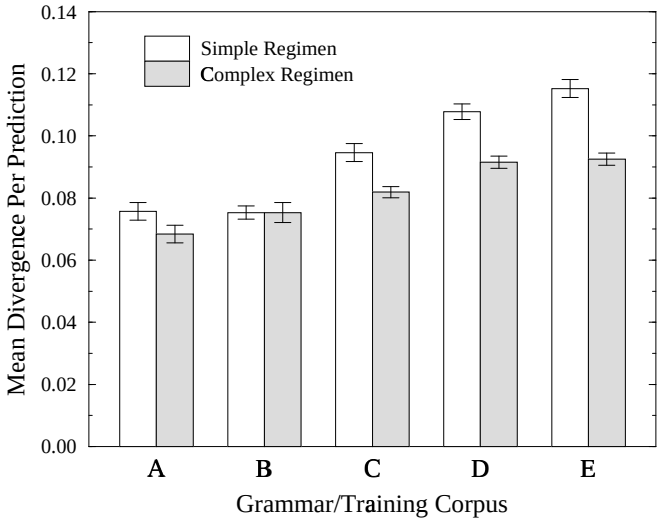

Figure 2: Final divergence error—note that lower values correspond to better performance. Means and standard error bars were computed for the best 16 of 20 trials.

diction distributions give the grammar and the sentence up to that point. Because the grammars are stochastic and context-free, these expectations are quite easy to generate.

Results and Discussion

Figure 2 shows the mean divergence error per word on the testing corpora, averaged over the 16 trials yielding the best performance in each condition. Overall, the complex training regimen yielded better performance than the simple regimen, $F(1,150)=53.8$, $p<.001$. Under no condition did the simple training regimen outperform the complex training regimen. Moreover, the advantage in starting complex increased with the proportion of fully constrained relative clauses, $F(4,150)=5.8$, $p<.001$. This conforms with the idea that starting small is most effective when important dependencies span uninformative clauses. Nevertheless, against expectations, starting small failed to improve performance even in class A where relative clauses do not conform to semantic constraints imposed by the preceding noun.

It is important to establish, however, that the network was able to master the task to a reasonable degree of proficiency in the complex regimen. Otherwise, it may be the case that none of the networks were truly able to learn. Average divergence error was 0.074 for networks trained on corpus A_{75} and 0.100 for networks trained on corpus E_{75}, compared with an initial error of 2.6. The class E languages are harder because semantic constraints force the network to make use of more information in predicting the contents of relative clauses. By way of anecdotal evidence, the networks appear to perform nearly perfectly on sentences with up to one relative clause and quite well on sentence with two relative clauses.

Figure 3 compares the output of a network trained exclusively on corpus E_{75} with the optimal outputs for that grammar. The behavior of the network is illustrated for the sentences Boy who chases girls who sing walks and Dogs who chase girls who sing walk. Note, in particular, the prediction of the main verb following sing. Predictions of this verb are not significantly degraded even after two embedded clauses. The network is clearly able to recall the number of the main noun and has a basic grasp of the different actions allowed on dogs and humans. It is, however, still unsure that boys are not allowed to bite and that dogs cannot sing. It also did not quite learn the rule that dogs cannot walk something else. Otherwise, the predictions are very close to optimal, including the fact that cats and humans cannot be walked.

For sentences with three or four clauses, such as Dog who dogs who boy who dogs bite walks bite chases cat who Mary feeds, performance of the networks was considerably worse. To be fair, however, humans are generally unable to parse such sentences without multiple readings. In addition, fewer than 5% of the sentences in the most complex corpora were over nine words long. This was necessary in order to match the average sentence-length statistics in Elman's corpora, but it did not provide the network sufficient exposure to such sentences for any hope of learning them. Interestingly, the networks were only about 4% worse on the testing set compared with the training set, indicating that they did not memorize the training sentences to a significant extent.

The best measure of network performance would appear to be a direct comparison with the results published by Elman (1991). However, there are problems with this approach. Because Elman did not use a standard form stochastic grammar, it was not possible to produce the theoretically correct predictions against which to rate the model. Instead, *empirically* derived probabilities given the sentence context were calculated. Presumably, these probabilities were compiled over replications in the testing set of the entire sentence context up to that point. Unfortunately, this type of empirically based language model tends to "memorize" the training corpus, particularly the long sentence contexts which are often unique.

Of the networks trained exclusively on corpus A_{75}, the one with median performance was selected for evaluation against an empirical language model trained on our A_{75} testing corpus. Elman reported a final error of 0.177 for his network (using, we believe, Minkowski-1 or city-block distance). Our selected network had an error of 0.485 against the model, which would seem to be considerably worse. However, city-block distance is not well-suited for probability distributions. Better measures are the mean cosine of the angle between target and output vectors, and their divergence. The selected network had an average cosine of 0.864, which is slightly better than the value of 0.852 that Elman reported.

However, comparison of the empirically derived predictions against the theoretically derived predictions, which represent the true desired behavior of the network, indicate that the former are actually quite poor. When evaluated against the theoretical predictions, the empirical model had a mean divergence of 1.897, a distance of 0.413, and a cosine of 0.881. In contrast, when compared against the same correct predic-

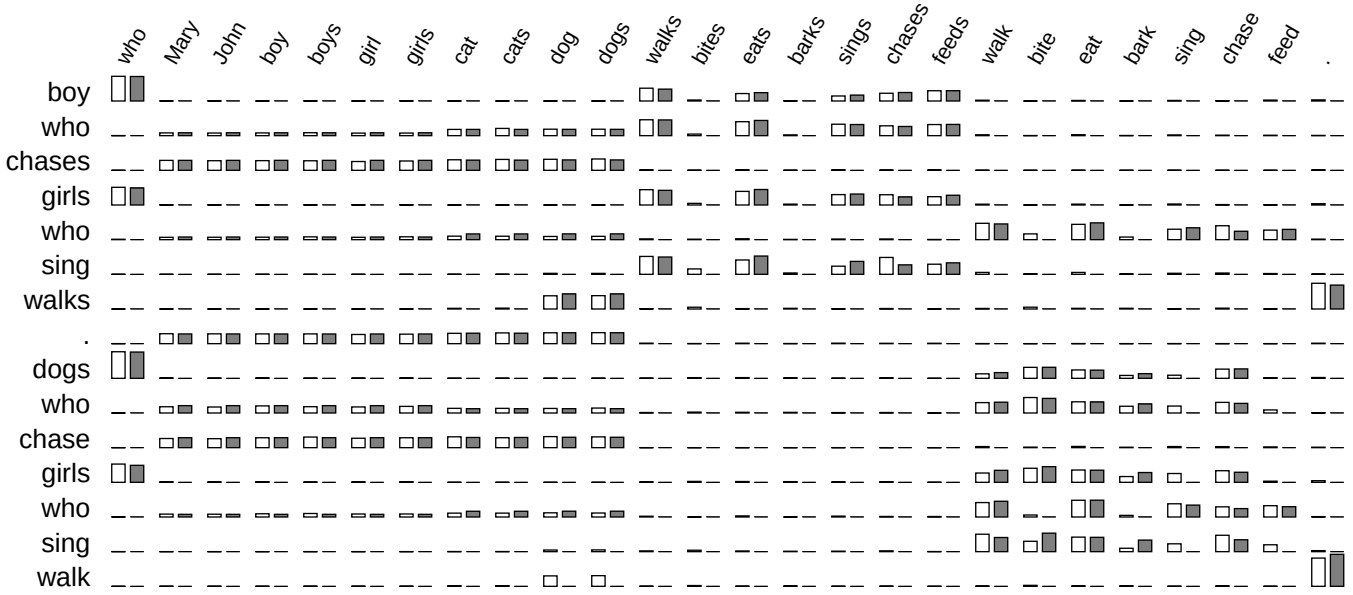

Figure 3: Predictions of a network on two sample sentences (white bars) compared with the optimal predictions given the grammar (filled bars). All values shown are the square root of the true values to enhance contrast.

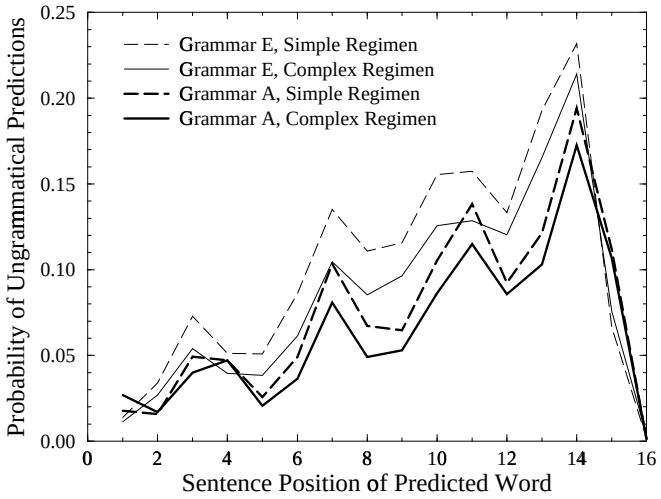

Figure 4: Strength of illegal (ungrammatical) predictions versus word position. Values are averaged over all 20 networks trained in each condition.

tions, the selected network had a divergence of 0.070, a distance of 0.158, and a cosine of 0.978. Thus, by all measures, the network's performance is better than that of the empirical model. In fact, an empirical model trained on 250,000 sentences generated by the A_{75} grammar, including the 10,000 sentences in the A_{75} testing corpus, did not even perform as well as the network against the theoretical predictions (divergence 0.902, distance 0.206, and cosine 0.944). Therefore, such an empirical model is not a good basis for evaluating the network or for comparing the network's behavior to that of Elman's network.

One possibility is that, although networks trained in the small regimen might have worse performance overall, they might nonetheless have learned long-distance dependencies better than networks trained the complex regimen. To test this hypothesis, we computed the total probability assigned by the network to predictions that could not, in fact, be the next word in the sentence, as a function of position in the sentence (see Figure 4). In general, fewer than 8 of the 26 words are legal at any point in a sentence produced by grammar E_{75}. Overall, performance declines with word position (except for position 16 which can only be end-of-sentence). However, even 17% of the total output activation spread over 18 illegal words is respectable, considering that randomized weights produce about 71% illegal predictions. More importantly, the complex-regimen networks outperform the simple-regimen networks irrespective of word position, $F(1,15)=25.7$, $p<.001$.

Although "starting small" failed to prove effective in the main experiments, we attempted to find conditions under which the simple training regimen would provide an advantage, in order to explain Elman's previous findings. First, we constructed additional corpora for which starting small might be expected to be beneficial: corpora composed entirely of complex sentences, and a sixth class of grammars (A') with no dependencies (including number agreement) between main and embedded clauses. However, the complex training regimen continued to yield slightly better performance than the simple regimen (mean divergence: 0.083 vs. 0.085 for A_{100}; 0.119 vs. 0.127 for E_{100}; 0.082 vs. 0.084 for A'_{75}, respectively). Another possibility was that the choice of training parameters was responsible for the effect. Therefore, networks were trained without momentum, without the use

of softmax constraints, and with a sum-squared error measure, rather than divergence. These parameters are identical to those chosen by Elman. Learning rates ranging from 1.0 to 0.0005 crossed with initial weight ranges from ± 1.0 to ± 0.0001 were evaluated. Under no conditions did networks trained with the simple regimen perform significantly better than networks trained with the complex regimen. However, with very small initial weights, a few of the networks using the complex regimen required nearly 15 epochs (about a million word presentations) to break through an early plateau in performance. Note, however, that such networks went on to achieve reasonable performance, although no networks trained under Elman's conditions performed as well as those trained with softmax and divergence error.

Conclusions

It is apparent that simple recurrent networks are able to learn quite well when trained exclusively on a language with only a small proportion of simple sentences. The benefit of starting small does not appear to be a robust phenomenon for languages of this type and starting small often proves to be a significant hindrance. It is not necessary to present simplified inputs to aid the network in learning short-term dependencies initially. Simple recurrent networks learn this way naturally, first extracting short-range correlations and building up to longer-range correlations one step at a time (see, e.g., Servan-Schreiber, Cleeremans & McClelland, 1991). Starting with simplified inputs allows the network to develop inefficient representations which must be restructured to handle new syntactic complexity.

An important aspects of Elman's (1993) findings was that a network was able to learn when the full range of data was presented initially and the network's memory was limited. Although the current work did not address this technique directly, Elman reported that networks trained with limited memory did not learn as effectively as those trained with simplified input. Given that, in the current work, we found that the simple training regimen was inferior to training on the full complex grammar from the outset, it seems unlikely that hindering the network's memory would be of any benefit.

It should be acknowledged, however, that there are situations in which starting with simplified inputs may be necessary. So-called "latching" tasks (Bengio, Simard & Frasconi, 1994; Lin, Horne & Giles, 1996) require networks to remember information for extended periods with no correlated inputs. Bengio and colleagues have argued that recurrent networks will have difficulty solving such problems because the propagated error signals decay exponentially. This is taken as theoretical evidence that an incremental learning strategy is more likely to converge (Giles & Omlin, 1995). However, such situations, in which dependencies span long, uninformative regions, are not at all representative of natural language.

Important contingencies in language and other natural time series problems tend to span regions of input which are themselves correlated with the contingent pair. In these cases, recurrent networks are able to leverage the weak short-range correlations to learn the stronger long-range correlations. Only in unnatural situations is it necessary to spoon-feed a network simplified input, and doing so may be harmful in most circumstances. The ability of such a simplified network model to learn a relatively complex prediction task leads one to conclude that it is quite plausible for a human infant to learn the structure of language despite a lack of negative evidence, despite experiencing unsimplified grammatical structures, and despite detailed, innate knowledge of language.

Acknowledgements

This research was supported by NIMH Grant MH47566 and an NSF Graduate Fellowship to the first author. We thank Jeff Elman, John Lafferty, and Jay McClelland for helpful discussions of issues related to this work. Correspondence may be sent either to Doug Rohde, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, dr+@cs.cmu.edu, or to David Plaut, Mellon Institute 115–CNBC, 4400 Forbes Avenue, Pittsburgh PA 15213–2683, plaut@cmu.edu.

References

Angluin, D. (1988). *Identifying languages from stochastic examples* (Tech. Rep. YALEU/DCS/RR-614). New Haven, CT: Yale University, Department of Computer Science.

Baker, C. L. (1979). Syntactic theory and the projection problem. *Linguistic Inquiry, 10*, 533–581.

Bengio, Y., Simard, P. & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks, 5*, 157–166.

Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning, 7*, 195–225.

Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. *Cognition, 48*, 71–99.

Giles, C. L. & Omlin, C. W. (1995). Learning, representation and synthesis of discrete dynamical systems in continuous recurrent neural networks. In *Proceedings of the IEEE Workshop on Architectures for Semiotic Modeling and Situation Analysis in Large Complex Systems*, Monterey, CA, August 27–29.

Gold, E. M. (1967). Language identification in the limit. *Information and Control, 10*, 447–474.

Hinton, G. E. (1989). Connectionist learning procedures. *Artificial Intelligence, 40*, 185–234.

Kutas, M. & Hillyard, S. A. (1980). Reading senseless sentences: Brain potentials reflect semantic incongruity. *Science, 207*, 203–205.

Lin, T., Horne, B. G., & Giles, C. L. (1996). *How embedded memory in recurrent neural network architectures helps learning long-term temporal dependencies* (Tech. Rep. CS-TR-3626, UMIACS-TR-96-28). College Park, MD: University of Maryland.

Luce, D. R. (1986). *Response times.* New York: Oxford.

McClelland, J. M. & O'Regan, J. K. (1981). Expectations increase the benefit derived from parafoveal visual information in reading words aloud. *Journal of Experimental Psychology: Human Perception and Performance, 7*, 634–644.

Neisser, U. (1967). *Cognitive psychology*. New York: Appleton-Century-Crofts.

Newport, E. L. (1990). Maturational constraints on language learning. *Cognitive Science, 14*, 11–28.

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.