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

Douglas L. T. Rohde (dr+@cs.cmu.edu)
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213–3890

David C. Plaut (plaut@cmu.edu)
Center for the Neural Basis of Cognition
Mellon Institute 115
4400 Forbes Avenue
Pittsburgh, PA 15213–2683

Abstract

Prediction is believed to be an important component of cognition, particularly in natural language processing. 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 in learning by limited cognitive resources (Elman, 1993, *Cognition*). The current work reports evidence that starting with simplified inputs is not necessary in training recurrent networks to learn pseudo-natural languages; in fact, delayed introduction of complex examples is often an impediment. We suggest that the structure of natural language can be learned without special teaching methods or limited cognitive resources.

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 (also see Marcus, 1993). Computational theory suggests that this is indeed problematic, as Gold (1967) has shown that, without negative examples, no superfinite class of languages is learnable, including the regular, context-free, and context-sensitive language classes. 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 incorrect predictions. Indeed, there is considerable empirical evidence that humans generate expectations in processing natural language and that these play an active role in comprehension (see, e.g., Neisser, 1967; Kutas & Hillyard, 1980; McClelland, 1988; McClelland & O'Regan, 1981).

Elman (1991, 1993) provided an explicit formulation of how a system might learn 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 having 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 initially restricted in its complexity (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 it had 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 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 the language from the start. Only under very contrived circumstances did starting with simple sentences reliably aid learning and, in most conditions, it was a hindrance. Furthermore, starting with the full language was of even 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 manipulation of the training regimen or internal memory is required to induce this property. Thus, the current work calls into question support for the claim that initially limited cognitive resources or other maturational

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	→	barks | sings | walks | bites | eats | bark | sing | walk | bite | eat
VT	→	chases | feeds | walks | bites | eats | chase | feed | walk | bite | eat

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.

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 verbs 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.

Grammars such as this are of interest because they 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 potentially long embeddings. Elman's grammar, however, was essentially purely syntactic, involving little or no 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 learned and 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 level of information a noun's modifying phrase contains about the identity of the noun 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 distribution, and hence to model the grammar, is what we expect the networks to learn.

Corpora

Cleeremans, Servan-Schreiber, and McClelland (1989) showed that a simple recurrent network, when trained to predict a finite-state language involving embedded structure, was aided when the embeddings were somewhat dependent on the surrounding context. In order to study this effect on our linguistic task, 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, 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 presumably the most English-like. Finally, a sixth class, N, was produced involving no semantic constraints, only number agreement, much like Elman's grammar.

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 24 grammars (six classes of semantic constraints crossed with 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, including a fully connected projection from "context" units whose activations are copied from hidden units at the previous time step. Each of the 26 input words was represented by a separate (localist) input unit. 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 sigmoid output units used by Elman. Specifically, the activation a_j of each output unit j was set to $\exp(x_j)/\sum_j \exp(x_j)$ where x_j is the total input to unit j. All other used the standard sigmoid activation function $(1 + exp(-x_j))^{-1}$. Error feedback was provide to the network in terms of the *divergence* (Hinton, 1989) between each

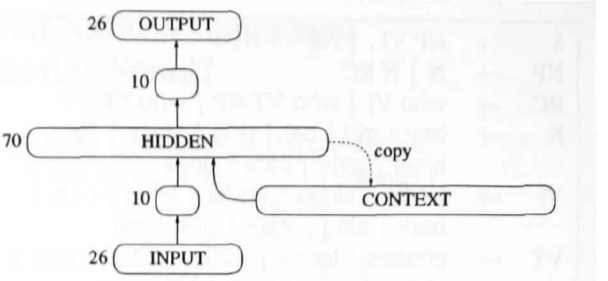

Figure 1: Network architecture. Each solid arrow represents full connectivity between layers (with numbers of units in parentheses). Hidden unit states are copied to corresponding context units (dashed arrow) after each word is processed.

output unit's target value t_j and its activation, $t_j \log(t_j/a_j)$. Note that when the target is 0, this value is by convention 0 as well. Therefore, error is injected only 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 back-propagated 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, sentence boundaries were indicated clearly by end-of-sentence markers.

Experiments

For each of the six language classes, two training regimens were carried out. In the *complex* regimen, the network was trained on the 75% complex corpus for 25 epochs with a fixed learning rate. The learning rate was then reduced 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 (0, 25, and 50% complex sentences). It was then trained on the fourth corpus (75% complex) for 10 epochs, followed by a final epoch at the reduced learning rate. The six extra epochs of training on the fourth corpus (not included in Elman's design) were included 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 learning rate of 0.004 (reduced to 0.0003), momentum of 0.9, 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 a learning rate of 0.1 (reduced to 0.06), no momentum, and initial weights in the ± 0.001 range; also, softmax constraints were not used and squared error was employed during training.

Both complex and simple trials were run for each of the six grammar classes. Twenty replications of each condition were performed, resulting in 240 total trials. Although the actual next word occurring in the sentence served as the target

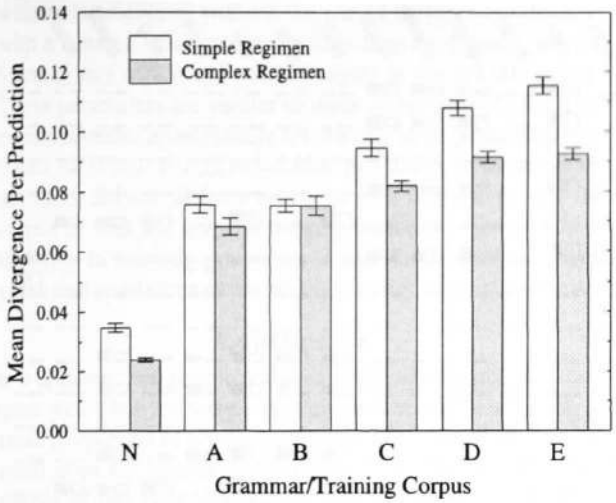

Figure 2: Final divergence error—note that lower values correspond to better performance. Means and standard error bars were computed for the best 16 of 20 trials.

output during training, the network was expected to produce a probability distribution over all possible words. The target vectors in the testing corpora consisted of the theoretically correct distributions give the grammar and the sentence up to that point. Because the grammars are stochastic and context-free, these expectations are straightforward 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 produced better performance than the simple regimen, $F(1,180)=72.8$, $p<.001$. 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 (A vs. E: $F(1,60)=8.55$, $p=.005$). 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 in which relative clauses are not semantically constrained by the head noun. Starting small was a particular hindrance on the purely syntactic grammar, N.

It is important to establish 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 0.068 for networks trained on corpus A_{75}, 0.092 on corpus E_{75}, and 0.024 on N_{75}, compared with an initial error of 2.6. Informally, the networks appear to perform nearly perfectly on sentences with up to one relative clause and quite well on sentences 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 to dogs and humans. It nearly mastered the rule that dogs cannot walk something else. It is, however, still unsure that boys do not bite and that dogs may bark, but not sing. Otherwise, the predictions appear to be nearly optimal.

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 was considerably worse. To be fair, however, humans have difficulty parsing 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 well. Additionally, the network, which was originally designed to learn the pure-syntax language, may have had too few hidden units to easily represent all the information necessary to process long, semantically-constrained sentences.

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 many sentences generated by the grammar. Unfortunately, this type of empirically based language model tends to "memorize" the often unique, long sentences in the training corpus and generalizes poorly.

We therefore trained an empirical model on the N_{75} testing corpus, as well as 240,000 additional sentences produced by the same grammar. Elman reported a final error of 0.177 for his network (using, we believe, Minkowski-1 or city-block distance). Our best 16 networks trained on the N_{75} corpus had an average error of 0.285 when evaluated against the model, which would seem to be considerably worse. However, city-block distance is not well-suited for probability distributions. A better measure (in addition to divergence) is the mean cosine of the angle between target and output vectors. The selected network had an average cosine of 0.929, which is somewhat better than the value of 0.852 that Elman reported.

Nevertheless, comparison of the empirically derived predictions against the theoretically derived predictions, which represent the true desired behavior of the network, indicate that the empirical predictions are actually quite poor. When evaluated against the theoretical predictions, the empirical model, which had been trained on 250,000 sentences, had a mean divergence of 1.086, a city-block distance of 0.246, and

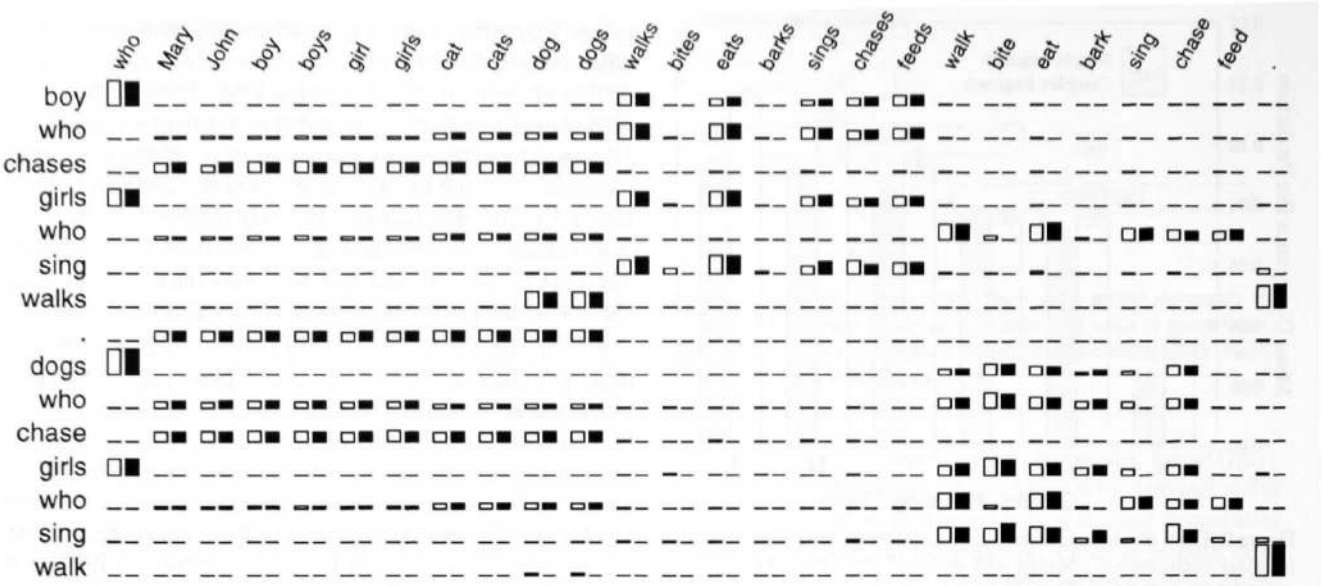

Figure 3: Predictions of a network on two sample sentences (white bars) compared with the optimal predictions given the grammar (filled bars). All values shown are the square root of the true values to enhance contrast.

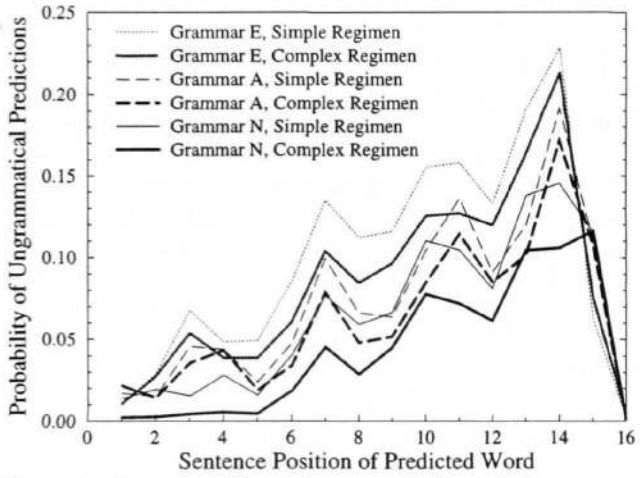

Figure 4: Strength of illegal (ungrammatical) predictions versus word position. Values are averaged over the best 16 of 20 networks trained in each condition.

a cosine of 0.934. In contrast, when compared against the same correct predictions, the networks had a divergence of 0.024, a distance of 0.088, and a cosine of 0.992. Thus, by all measures, the network's performance is better than that of the empirical model. Therefore, such a 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 may 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 (ungrammatical predictions increase) with word position, except for position 16 which can only be end-of-sentence. However, even 21% 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 at each sentence position between 5–14 (typically involving embeddings; $F(1,15)>4.31$, $p\leq.031$, for each position).

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. First, we constructed conditions for which one might expect starting small to be beneficial: a sixth class of grammars, A′, with no dependencies between main and embedded clauses (including number agreement), and corpora composed entirely of complex sentences. However, the complex training regimen continued to yield equivalent performance to the simple regimen (mean divergence: 0.079 vs. 0.080 for A'_{75}, $F(1,30)=0.135$, $p=0.716$; 0.078 vs. 0.081 for A_{100}, $F(1,22)=1.14$, $p=0.298$; 0.112 vs. 0.120 for E_{100}, $F(1,22)=1.46$, $p=0.241$). Only in the extreme case of A'_{100} did starting small yield a significant benefit (0.105 complex vs. 0.064 simple, $F(1,22)=6.99$, $p=0.015$).

A remaining possibility is that the difference in training parameters, or slight differences in corpora, between our experiments and Elman's were responsible for our discrepant results. Therefore, we eliminated all known differences between grammar class N and Elman's and trained networks

without momentum, without the use of softmax constraints, with a squared error measure, rather than divergence, with a learning rate of 0.1 and initial weights in the ± 0.001 range. These parameters are similar to those chosen by Elman. The results revealed an advantage for starting large (squared error: 0.088 vs. 0.107, $F(1,22)=246.986$, $p<0.001$), however these networks did not perform nearly as well as those trained on corpus N with the original training methods (squared error: 0.0042). In learning grammars similar to Elman's, we have yet to find conditions under which starting small is beneficial.

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 aspect 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 the simple training regimen inferior to training on the full complex grammar from the outset, it is unlikely that hindering the network's memory would be of any benefit. Indeed, preliminary results not reported here seem to bear out this prediction.

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 information. 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 train a network initially on 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, Jay McClelland, the CMU PDP research group, and three anonymous reviewers for helpful comments and/or discussions.

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.

Cleeremans, A., Servan-Schreiber D., & McClelland J. (1989). Finite state automata and simple recurrent networks. *Neural Computation, 1*, 372–381.

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.

Marcus, G. F. (1993). Negative evidence in language acquisition. *Cognition, 46*, 53–85.

McClelland, J. L. (1988). Connectionist models and psychological evidence. *Journal of Memory and Language, 27*, 107–123.

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.