Sequential Connectionist Networks for Answering Simple Questions about a Microworld

Robert B. Allen
Bell Communications Research

Sequential back-propagation networks were trained to answer simple questions about objects in a microworld. The networks transferred the ability to answer this type of question to patterns on which they had not been trained. Moreover, the networks were shown to have developed expectations about the objects even when they were not present in the microworld. A variety of architectures were tested using this paradigm and the addition of channel-specific hidden layers was found to improve performance. Overall, these results are directed to the approach of building language users with connectionist networks, rather than language processors.

Introduction

Because neural algorithms such as back-propagation [9] are such effective techniques for machine learning, it is now possible to seriously consider developing systems which learn to *use* language. Moreover, neural networks have many other characteristics which make them especially suitable for such an effort. They are sensitive to context, they can adapt to exceptions, and input from diverse sources can be easily combined. The concept of developing a language user is an alternative to the usual approach in artificial intelligence of attempting to process the components of language and then to synthesize an "understanding" from those components. Rather, the approach suggested here is to train networks under a broad enough range of conditions to be able to understand and respond with language-like stimuli. Potentially, this approach may provide a robust basis for linguistic processes ranging from translation [1] to speech recognition. Of course, the extent to which these networks can be said to actually 'have' language may well be as difficult and controversial as the evaluation of linguistic capabilities of apes (see [7]).

In the research described here, sequential networks were trained to accept language-like stimuli which refer to objects in the microworld. This paradigm is based on the assumptions that language is most readily acquired through interaction with the world [1] and that language learning is essentially a supervised learning process. Inputs of two types were employed, a coded microworld and sequential verbal codes, which formed questions about the microworld. One type of sequential network which might be applied to this task is shown on the left of Fig. 1. This network, which may be termed an output-feedback network, was developed by Jordan [6] to model articulation. The output units from one cycle feed 'state' units which are used as extended inputs for later cycles. As shown on the right side of Fig. 1 a variation of that procedure, suggested by Elman [4], draws feedback from the hidden layer rather than the output layer. Fig. 2 presents a sequential network which may be termed a generalized hierarchical-sequential network. In this network inputs are divided into separate channels and have extra hidden layers for each of the sets of input units, as well as for the state units. This architecture has the advantage of being modular, hence it might be readily adapted to multi-processor computers and perhaps specialized perceptual hardware.

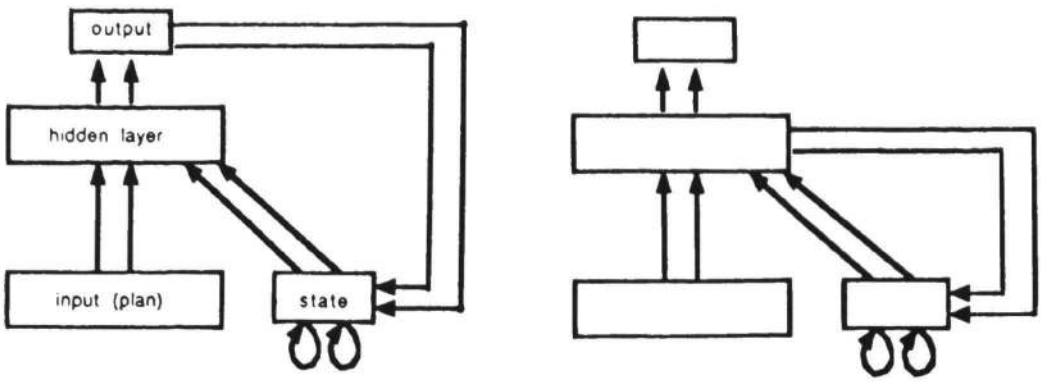

Fig. 1. Simple sequential networks with feedback from output and hidden units.

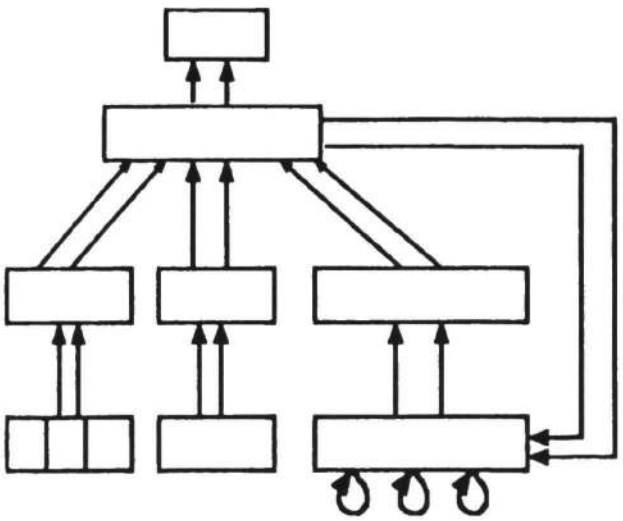

Fig. 2. Generalized hierarchical-sequential network.

Procedure

Coding

The verbal inputs were composed from questions from a vocabulary of 27 terms which were coded with randomly assigned 6-bit -1/1 codes. There were 16 output terms and their coding was randomly selected from 5-bit 0/1 codes. One set of inputs presented static codes which were characterized as objects in a 'perceptual' field. The perceptual field was composed of 3 slots, each slot consisting of 5 bits, in which any one of 8 objects could appear. 3 of these bits encoded the objects themselves and two additional bits coded features. The objects themselves were coded with randomly selected -1/1 codes, while empty slots were filled with nulls. In addition to the 3 bits which uniquely specified the objects, two additional bits were correlated with each object in the proportions shown in Table 1. While the probabilistic features will be more difficult to learn than perfectly correlated features, they are necessary to guarantee that the network attends to the microworld. On one hand, these bits might be considered to be an explicit feature

such as color. For instance, it would be possible to say that **object$_1$** had **color$_1$**. Alternatively, these bits could be thought of as context bits. For instance, some objects are highly correlated with places, while other objects are less likely to be correlated. The additional bits were associated with the objects in the following proportions:

object	feature	
	A	B
1	.9	.9
2	.8	.9
3	.7	.9
4	.6	.9
5	.4	.1
6	.3	.1
7	.2	.1
8	.1	.1

Table 1. Probability of object tokens having features A and B.

Question Construction and Sequential Presentation

The sequential presentation of verbal information is illustrated in Table 2. In the example two objects are in the perceptual field, **object$_2$** in **slot$_1$** and **object$_1$** in **slot$_2$**. Across the four intervals, the coded forms of the words **is**, **this**, **object$_1$**, and **fA$_1$** are presented. Because the input is sequential, correct output responses were generally not known until the question was complete. Before the correct output is known error values of zero were back-propagated. (A similar procedure in which there was no back-propagation on those trials produced similar, but slightly worse results.) These cycles are indicated by *** in the table and termed *don't care cycles* after the *don't care units* of Jordan.

interval	perceptual input			verbal input	verbal output
1	$o_2/fA_2/fB_1$	$o_1/fA_1/fB_1$	null	is	***
2	$o_2/fA_2/fB_1$	$o_1/fA_1/fB_1$	null	this	***
3	$o_2/fA_2/fB_1$	$o_1/fA_1/fB_1$	null	$object_1$	***
4	$o_2/fA_2/fB_1$	$o_1/fA_1/fB_1$	null	fA_1	yes

Table 2. Sequences of codes for typical question answering procedure.

Input/output patterns were prepared from the templates shown in Table 3. With the exception of a few very simple commands (e.g., repeat, describe), the input patterns were questions. A question such as **What fA is the objX$_1$** might be read, with nouns inserted, as **What color is the car?** While the answer, **fAX$_1$**, might be read **blue**. Clearly, the templates are somewhat ad hoc and stylized. The questions were associated with appropriate perceptual inputs; thus, the 8 objects could appear in 3 slots of the perceptual field. In cases with two objects in the perceptual field and only one of them was necessary to answer the question, the second object was randomly selected and its features were constrained so as not to conflict with the question. When questions could be answered yes/no, equal numbers of yes and no questions were prepared. 3242 unique inputs were generated, and from this set 25 were randomly chosen for

transfer.

verbal input	verbal output
object absent	
repeat $objX_1$	$objX_1$
repeat fAX_1	fAX_1
repeat fBX_1	fBX_1
one object present	
repeat $objX_1$	$objX_1$
repeat fAX_1	fAX_1
repeat fBX_1	fBX_1
what do you see	$objX_1$
describe what you see	$objX_1$
is this the $objX_1$	yes/no
do you see the $objX_1$	yes/no
is this $objX_1$ fAX_1	yes/no
is this $objX_1$ fBX_1	yes/no
what fA is the $objX_1$	fAX_1
what fB is the $objX_1$	fBX_1
what is the fA of the $objX_1$	fAX_1
what is the fB of the $objX_1$	fBX_1
is the $objX_1$ in the slotS1	yes/no
two objects present	
is the $objX_1$ fAX_1	yes/no
is the $objX_1$ fBX_1	yes/no
which is fAX_1	$objX_1$
which is fBX_1	$objX_1$
what fA is the $objX_1$	fAX_1
what fB is the $objX_1$	fBX_1
what is the fA of the $objX_1$	fAX_1
what is the fB of the $objX_1$	fBX_1
is the $objX_1$ in the slotS1	yes/no
is the $objX_1$ over/under the $objX_2$	yes/no
which is over/under the $objX_1$	$objX_2$

Table 3. Input/output templates.

Network Parameters

The weights from the hidden units to the state units were fixed at 1.0 and the self-weights on the state units were 0.5. All of the other weights were adaptive with η=0.01 and α=0.9. The networks described below were trained for 200K pattern presentations. Except as noted below, the simple networks had 15 perceptual units, 6 verbal input units, 50 state units, 50 hidden units, and 5 output units. In addition, the hierarchical networks had 15, 6, and 50 units in the perceptual, verbal, and state-hidden layers respectively. At the beginning of each question the state units were reset to zero; tests demonstrated that learning occurred without reset, although it was faster and better with resets.

Results

The transfer set consisted of 25 questions and because each question required a one word (5 bit) response, a total of 125 bits had to be generated. The sequential network with feedback taken from the output (left side of Fig. 1) made errors on 29 bits (12 words). The hidden layer feedback network (right side of Fig. 1) performed somewhat better, with 23 bit errors and 9 word errors. Indeed, more complex networks (e.g., the PVS, see Table 4) made as few as 6 bit errors and 2 word errors (see below). It is perhaps remarkable that these networks can learn this task because they rarely get explicit training for storing words in the early part of the sentence. For instance, in questions such as the one shown in Table 2, the network has to remember that the question concerns **object$_1$**.

Architecture Manipulation

The hierarchical-sequential network (Fig. 2) can be thought of as a family of networks which may be tested separately. The errors for the 8 possible networks (formed by all possible combinations of the presence/absence of the perceptual, verbal, and state hidden layers) is shown in Table 4. As a short notation, these networks may be referred to with three-letter codes for instance, a PNS network would have a perceptual hidden layer, no verbal hidden layer, and a state-hidden layer. The NNN network is the network shown at the right side of Fig. 1. All of the other networks perform better than the NNN network, and the best is the PVS network.

	N		S	
	N	P	N	P
N	23(9)	7(4)	17(8)	12(6)
V	18(6)	14(5)	14(7)	6(2)

Table 4. Bit errors (word errors) for different networks.

Additional tests showed that these results replicated, essentially following the pattern in Table 4. However, with other data sets the PVS network is not consistently found to be best. Moreover, considerable caution must be exercised in comparing the different architectures in Table 4 because they include different numbers of neurons and weights. As a control for this, several other networks were tested. First, an output feedback network (left side of Fig. 1) with 100 hidden units was tested; this performed relatively poorly, 26(13). As a second test, a NNN network with 65 units in both the hidden and state layers was tested, this had 13 bit errors and 9 word errors.

Several other architectures, related to hierarchical-sequential network described above, may also be considered. For instance, a network was investigated in which the state units and a state hidden layer were attached to the verbal hidden layer. With 30 hidden units, 20 verbal hidden units, 20 state units, and 20 state-hidden units, this network performed the task with 11 bit errors and 6 word errors.

Semantic Memory and Categorization

Previous connectionist research on semantic memory [5, also Rumelhart unpublished] may be extended by considering semantic memory in this paradigm with explicit verbal training. Thus, the PVS network trained above was presented with the question **What fA/B is the object$_1$?** when the object was absent from the perceptual field. The decoded responses are shown in Table 5, where 1 or 2 indicates one of the features belonging to that feature type, and X indicates an apparently meaningless answer. For 7 of the

objects the network learned the correct values of **fB**, which is more consistently associated with the objects than **fA** (see Table 1). However for **fA**, the network consistently assumed that all objects had one feature value with the exception of $\mathbf{object_1}$, which caused an error.

object	feature	
	A	B
1	X	X
2	2	1
3	2	1
4	2	1
5	2	2
6	2	2
7	2	2
8	2	2

Table 5. Responses to feature questions with the object absent.

Additional tests have demonstrated that networks can learn about features although the features are never presented in the microworld. Moreover, the networks have been found to readily learn category names for grouping objects. On the other hand, a network in which the microworld was entirely absent showed only a small improvement above chance performance.

An Extended Generalization Test

When working with complex stimuli such as the questions and microworlds used here it is possible to consider generalization at many levels. While the transfer test described above consisted of randomly selecting test cases from a corpus in which the same question was often asked about several different configurations of the perceptual field, the test described here completely dropped training on one question and then tested that question during transfer. Specifically all 41 questions were removed from the corpus which asked whether $\mathbf{object_1}$ had $\mathbf{fA_1}$. Although other questions asked whether $\mathbf{object_1}$ had $\mathbf{fB_2}$ and whether other objects had $\mathbf{fB_1}$. A PVS network trained on the patterns which were not deleted, transferred quite well with 6 bit errors and 4 word errors.

Discussion

Networks were shown to learn and transfer the ability to answer questions in which coded 'verbal' questions and objects are presented sequentially. Moreover, evidence was presented for the utility of hierarchical-sequential networks. Naturally, this research may be extended in many ways. For instance in the sequential verbal input paradigm, linguistic constructs such as plurals or negation could readily incorporated. Indeed [8] reports the comprehension of pronouns which refer to objects in the microworld. In addition, the contribution of the hidden layers might be investigated through either analysis of the activations or parametric manipulation of the numbers of units. Presumably the additional hidden layers in the hierarchical network transform the input encoding to an encoding which is more easily intergrated with the other sources of information. In the case of the perceptual inputs this suggests that language can affect perception, in other words a type of *linguistic relativity*.

The research described here has focused on the task of answering questions as a means of generating feedback for language training. While this may seem restrictive at first, it is possible to imagine many variations in which the training would be less explicit. For instance, the questions might not be directed

to the network. If the question were posed to a different agent which made the response, the agent which is acquiring language might learn pairings of input and output from observation and perhaps imitation of the other agent. Furthermore, aside from answers to explicit questions there are many types of feedback for language use such as correctly completing a verbal command or instruction.

Finally, the strategy of developing language users which interact with a microworld may be extended beyond sequential verbal inputs. Additional work is underway in which verbal and microworld information is combined to produce language-like behavior, for instance networks which generate sequential outputs and manipulate the microworld [2] and multiple networks which communicate to complete tasks [3].

REFERENCES

1. Allen, R.B. Several studies on back-propagation and natural language. *Proceedings of the International Conference on Neural Networks* (San Diego, 1987), II/335-II/341.
2. Allen, R.B. Generation of verbal descriptions and action sequences with connectionist networks. submitted.
3. Allen, R.B. and Riecken, M.E. Interacting and communicating connectionist agents. *Proceedings of the International Neural Network Society* (Boston, 1988).
4. Elman, J.L. Finding structure in time. UCSD-CRL TR#8801.
5. Hinton, G. Learning distributed representations of concepts. *Proceedings of the Cognitive Science Society* (Amherst, MA, 1986), 1-12.
6. Jordan, M.I. Attractor dynamics and parallelism in a connectionist sequential machine. *Proceedings of the Cognitive Science Society* (Amherst, MA, 1986), 531-546.
7. Premack, D. *Gavagai!* (MIT Press, Cambridge, MA), 1986.
8. Riecken, M.E. and Allen, R.B. Anaphora and reference in connectionist language users. submitted.
9. Rumelhart, D.E., Hinton, G.E., and Williams, R.J., Learning internal representations by error propagation. In: D.E. Rumelhart and J.L. McClelland (Eds.), *Parallel Distributed Processing*. (vol. 2). 1986, 318-362.