Representing Variable Information with Simple Recurrent Networks

Catherine L. Harris
Jeffrey L. Elman

Department of Cognitive Science
University of California, San Diego

ABSTRACT

How might simple recurrent networks represent co-occurrence relationships such as those holding between a script setting (e.g., "clothing store") and a script item ("shirt") or those that specify the feature match between the gender of a pronoun and its antecedent? These issues were investigated by training a simple recurrent network to predict the successive items in various instantiations of a script. The network readily learned the script in that it performed flawlessly on the non-variable items and only activated the correct type of role filler in the variable slots. However, its ability to activate the target filler depended on the recency of the last script variable. The network's representation of the script can be viewed as a trajectory through multi-dimensional state space. Different versions of the script are represented as variations of the trajectory. This perspective suggests a new conception of how networks might represent a long-distance binding between two items. The binding must be seen as not existing between an antecedent and a target, but between a target item and the current global state.

INTRODUCTION

Researchers interested in how networks might be used to model aspects of natural language have begun to explore the representational capacities of simple recurrent networks (Elman, 1988; Servan-Schreiber, Cleeremans, & McClelland, 1988). These networks are chosen for study because they require minimal assumptions about the structure of language data. If words of a language are represented as random binary strings presented one at a time to a network, then sentences are simply temporal sequences of words. Elman (1988; 1989) has shown that even as unassuming a task as that of predicting the next word in the sequence can result in the extraction of interesting regularities. Although the simplicity of SRN's (simple recurrent networks) places limits on their usefulness as realistic models of natural language, it is hoped that some of the principles governing their behavior will hold for more complex architectures, or architectures which include SRN's as sub-components.

The SRN used in the current work has three layers. On each time step, the pattern of activation on the middle ("hidden-unit") layer is *copied* to another bank of units called the context layer. On the next cycle, the context layer together with the input layer feeds the hidden units. Only the forward-feeding connections are modified during training. Although the SRN has immediate access to only the current word and to the preceding state, past work has shown that the hidden layer will often come to represent a condensed, prediction-relevant record of past items.

Previous successes with SRN's suggest two further avenues of investigation:

- It has been shown that SRN's can, with the right training environment, develop internal representations for a word which are sensitive to the word's sentential context. Under what circumstances (what types of patterns and training environments) would a network be motivated to color the representation of an item with aspects of the larger context, such as its position in a discourse?
- One feature of natural languages is that regularities exist between non-adjacent items. Examples are subject-verb agreement, the gender/number match between pronouns and their antecedents, and the relationship between script setting (e.g, "restaurant") and script entities ("waitress," "menu"). How do recurrent networks fare at extracting these long-distance relationships, and what representations do they construct in doing so?

The two questions do not have to be addressed together, but there are a number of reasons for exploring them at the same time. One is convenience: a script may be composed of sentences embodying a variety of linguistic regularities, some of which can include the long-distance relationships we are interested in. An additional reason is that some long-distance relationships, like the relationship between the gender of a pronoun and the gender of its proper-name antecedent, span sentence boundaries and thus are most naturally explored in the context of connected sentences.

INSTANTIATION OF SCRIPT VARIABLES

Connectionist models of language have yet to seriously confront the problem of how networks might represent variable bindings (Norman, 1986). Bindings such as those between pronouns and their world-referents require a mapping from one system (linguistic symbols) to another system (a representation of the referent world). The mapping between the name "Mary" and a specific individual in the world is a between-system regularity. But regularities also exist within a single representational system. For example, in the surface form of words, the gender of a name agrees with the gender of the pronoun.

By definition, SRN's can only capture regularities that exist within a single representational system. It is thus important to stress that they are not an adequate vehicle for exploring all of the complexities of the binding problem. Nevertheless, one important step is to understand what sequential co-occurrence regularities SRN's can capture and what representations they will construct in doing so.

The Script Skeleton

We began our exploration of the representation of script information and long-distance relationships by constructing the simplest type of script: a fixed sequence of items, where an item can be either a constant (an invariant item) or a variable. We used the following script skeleton:

person1 asked *person2* if *subj.pronoun2* wanted to go to a *place*
at the *place subj.pronoun2* saw a *thing1 subj.pronoun2* liked
subj.pronoun2 showed *person1* the *thing1* and asked *obj.pronoun1* if *subj.pronoun1* liked it
subj.pronoun1 told *person2 subj.pronoun1* liked the *thing1* but *subj.pronoun1* wanted to get a *thing2*

Selecting *Laura* and *Ralph* as the characters, and *restaurant* as the script yields:

Laura asked Ralph if he wanted to go to a restaurant
At the restaurant he saw a sandwich he liked
He showed Laura the sandwich and asked her if she liked it
She told Ralph she liked the sandwich but she wanted to get a salad

Four sets of script variables and 6 proper names were used to generate instantiations of the script skeleton.

Place	*Thing1*	*Thing2*
restaurant	sandwich	salad
furniture store	sofa	chair
clothing store	shirt	jacket
record store	record	tape

3 male names: Ralph, John, Jeff
3 female names: Mary, Sue, Laura

The variables, sixteen constant items, an end-of-sentence marker and an end-of-script marker added up to 40 lexical items. Items were represented as unique bits in a 40-bit vector, meaning that input and output layers contained 40 units. Scripts were constrained so that each passage contained a male and a female character, although which occurred first varied. All possible combinations of 4 scripts, 9 male-female combinations, and 2 orders (female first name or male first name) resulted in 72 different scripts. The 72 scripts were concatenated in a random order and joined to form a sequence of 3780 items. This sequence constituted the training set.[1]

Weights were changed after each presentation of an input-output pair. Training was stopped when the decrease in error was negligible. With a low learning rate (.01), this required about 600,000 presentations of an input-output pair.

NETWORK PERFORMANCE

Two aspects of network performance were examined.

- How well does the network solve the task of predicting the next word when the word is a variable compared to when the word is a constant?
- What internal representations are constructed in the service of this prediction task?

Ability to Activate the Correct Targets

Network performance on one instantiation of a script appears in Table 1. At each step in the sequence, the target output is shown, followed by all output units activated at greater than 0.20 ("threshold"). Blanks after a target word mean that no output reached threshold on this cycle. Activation values have been multiplied by 100.

[1] One idiosyncrasy is that three of the *place* names are composed of two words, while one ("restaurant") is a single word. This means that instantiations of the restaurant script will yield sequences of length 51 instead of 53. A second idiosyncrasy is that in the "record store" script, the name of thing1 is a repetition of a token in the place name.

TABLE 1: Activations for One Script Instantiation

	Target Output	Activated Outputs Units		Target Output	Activated Output Units
1	*Laura*		27	*sandwich*	24
2	asked	asked 93	28	and	and 90
3	*Ralph*		29	asked	95
4	if	if 93	30	**her**	**him** 46, **her** 42
5	**he**	**he** 70, **she** 23	31	if	if 94
6	wanted	wanted 92	32	**she**	**he** 26, **she** 75
7	to	to 97	33	liked	liked 93
8	go	go 86	34	it	it 89
9	to	to 96	35	end1	end1 98
10	a	a 93	36	**she**	**he** 50, **she** 47
11	*restaurant*		37	told	told 91
12	end1	end1 89	38	*Ralph*	
13	at	at 79, end2 25	39	**she**	**she** 80
14	the	the 88	40	liked	saw 20, liked 80
15	*restaurant*	*restaurant* 20	41	the	the 90
16	**he**	**he** 43, **she** 50	42	*sandwich*	*sandwich* 28, *sofa* 23, *shirt* 37
17	saw	saw 76, liked 24	43	but	but 85
18	a	a 97	44	**she**	**he** 48, **she** 31
19	*sandwich*	*sandwich* 21	45	wanted	wanted 97
20	**he**	**he** 56, **she** 32	46	to	to 97
21	liked	liked 91	47	get	go 31, get 73
22	end1	end1 88	48	a	a 96
23	**he**	**he** 72, **she** 41	49	*salad*	
24	showed	showed 91	50	end1	end1 93
25	*Laura*		51	end2	end2 87
26	the	the 97			

Activating constants. For all but four of the constant slots (slots 13, 17, 40, 47), only the target output unit was activated at greater than threshold. Note how the similarity in the sequences "*he/she* wanted to go/get" caused competition between two outputs in script position 47.

Activating proper names and script-entities. No proper names were activated in the script instantiation in Table 2 or in the rest of the corpus. The network was consistently mute at these slots in the script. Inspecting Table 1 might lead one to believe that the network did have some ability to activate the appropriate script entities. For example, *restaurant* and *sandwich* appear to be appropriately activated in 15 and 19. Sampling over a complete run through the 72 scripts of the training set revealed that the network always activated the appropriate category of filler (e.g., one or more of the *thing1* variables for a *thing1* slot). However, targets were not more reliably

activated than non-targets.[2]

The difficulty in activating any class members for name slots, and in activating the target class member in the script variable slots, may be linked to the frequency of occurrence of the target items in the training set. A given proper name only occurred twice in one out of three instantiations, while *thing1* variables occurred slightly more frequently (three times in one out of four instantiations). The pronouns, on the other hand, were highly frequent variables in this training set. "He" and "she" each occurred four times per script instantiation. The network was more successful at activating pronouns than other variables. If an item appears infrequently during training, there will be few opportunities for the network to learn how to adjust the weights that will turn on the target unit in the output layer.

Activating pronouns. Comparing the relative activation of target and non-target pronouns in Table 1 shows that the network's ability to activate the target pronoun is limited. Pronouns occur in nine different script positions. Sampling over a complete run through the 72 scripts of the training set, it was found that in only three of these nine script positions (slots 5, 23, 39) was the network able to activate the target pronoun significantly better than the non-target pronoun.

Ignoring the script entities, the basic task of the network is to learn two variations on the script. In one variation, a male name occurs in the first name slot, dictating that the first four pronouns will be female and the remaining five pronouns slots will be male; in the second variation a female name occurs first. The network appears to need constant reminders of which of these two tracks it is on. For example, the reminder for the pronoun in position 44 is the pronoun in 39. What distinguishes the pronouns in slots 5, 23 and 39? The most recent reminder is either the current input (as in 39), or the immediately preceding item (5, 23).

Hidden Unit Analysis

A hierarchical cluster diagram of the 40 lexical items appears in Figure 1. Members of each variable class (*place, thing1, thing2*) are clustered together, suggesting that the network has extracted these type categories. (*Restaurant* may be grouped separately from the other place names because it is a one-word place name, rather than a two-word place name.) The male and female names are also clustered into separate categories, as one would expect if the network has discovered a correlation between these names and the values of pronoun slots. The tree's lack of balance and bushiness indicates that a number of items did not cluster into categories. For example, the two articles ("a" and "the") occurred in similar script positions, but were not similar to other items in the set. In general, the grammatical function words had such idiosyncratic environments that no abstract type could be extracted for them.

To better understand how the network carried information about pronoun identity forward from a cue like a proper name, we wanted a method of following the network's change of state over time. This was accomplished by reducing the dimensionality of the hidden unit layer by principal components analysis. Each input item activates a vector of 40 hidden units. By using the first two principal components of this vector, the state of the hidden-unit vector can be

[2] Why would *restaurant* and *sandwich* be actitvated regardless of what *place* name occurred in slot 11? In slot 42, why was *shirt* always most highly activated? One hypothesis is that at some point in training, these were the correct outputs, and the network slid down the error slope into a local minimum from which it was never able to emerge.

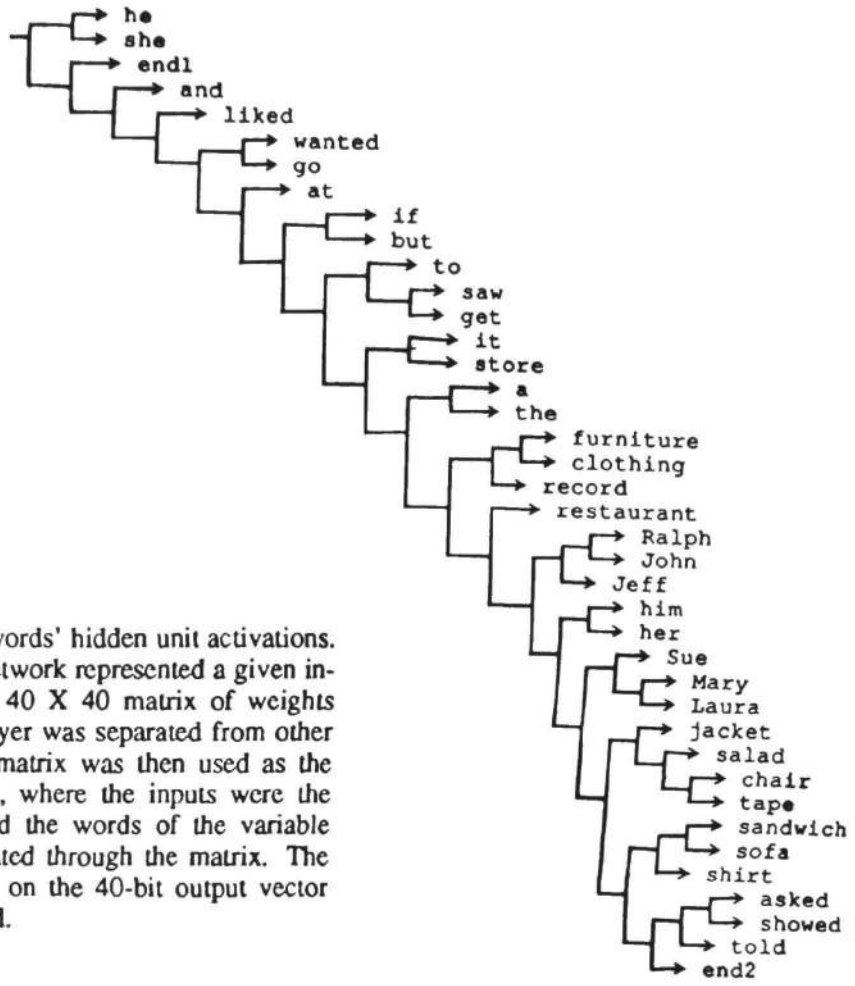

FIGURE 1: Cluster diagram of words' hidden unit activations. In order to determine how the network represented a given input irrespective of context, the 40 X 40 matrix of weights between the input and hidden layer was separated from other weights in the network. This matrix was then used as the weights in a two-layer network, where the inputs were the 40-bit vectors which represented the words of the variable classes. Activation was propagated through the matrix. The patterns of activation appearing on the 40-bit output vector were then hierarchically clustered.

approximated with a single point in a 2-D plot.

Figure 2 illustrates the state change in two versions of the first sentence of the skeletal script. The two versions are the two types of gender ordering, script setting held constant.

Following the trajectories from the starting point of "Laura" and "Ralph" one sees that the two versions are distinct early in the sentence. The difference in their paths becomes most marked immediately after getting the cue to gender ("Mary/John") and continues to be distinct through the first pronoun. The paths begin to come together after the fixed sequence "...wanted to go to a..." By the beginning of the next sentence (not depicted) the trajectories are identical. The paths diverge after receiving either "he" or "she" as input, collapse when "record" is encountered, and deviate again at the next pronoun. This good differentiation continues into the next sentence, which begins with a pronoun -- notably, one of the slots where the network was successful at activating the target pronoun (slot 23).

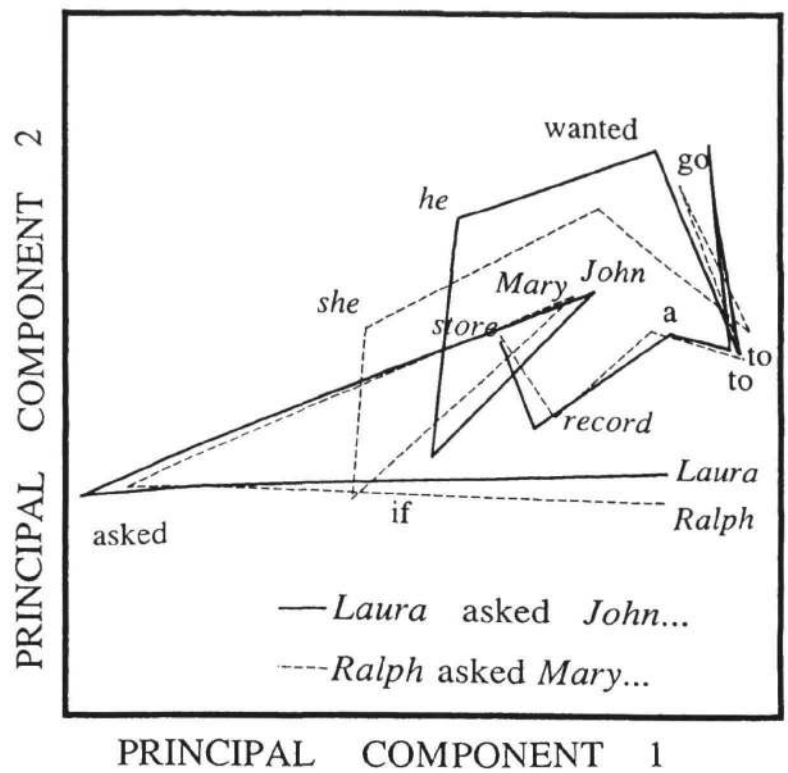

FIGURE 2: State change diagram for two versions of script sentence one. For a given state, labels indicate what word was the target. When versions had identical targets, only one label is given.

DISCUSSION

Studying how SRN's learn a skeletal script has allowed us to make the following observations about the properties of these networks:

- SRN's are good at extracting those distributional regularities which signal class membership. The current network did this even though the only clue to class membership was through co-occurrence relations with non-adjacent items. For example, the network learned that *thing2* variables form a class (Figure 3) even though the cue to class membership occurred 7 items previously.
- Although the network was able to construct long-term representations of the structure of the input, its ability to carry transient information about the value of a variable was limited. Information about the identity of a variable must be carried through the intervening items since the last cue. This information can be viewed as a path through state space. Information that the upcoming pronoun is to be a "he" will take the form of a slightly different trajectory than information that the upcoming pronoun is a "she."

To the extent that we want to view the rec net as having the ability to represent a long-distance relationship, we need to view the relationship as holding not between a temporally separated target and antecedent, but between the target and the current global state. This view is an intriguing one and merits further exploration. Nevertheless, the transiency of these relationships is sobering. It is difficult to imagine that speakers' ability to select the correct pronoun might be

mediated by a mechanism which requires all intervening states to be flavored with information about the identity of the upcoming pronoun. On the other hand, within-system regularities may exist, such as subject-verb agreement, which might prove more amenable to modeling with a device capable of capturing only fleeting relationships (Elman, 1989).

Work in progress extends these preliminary findings in two ways:

- We examine what factors will encourage a network to construct a representation of items which is sensitive to script position. For example, for many stereotyped scripts, it is likely that generic verbs ("like," "see") and function words will not be sensitive to script position, while items which help define script structure ("menu," "eat," "pay") will acquire position-sensitive representations.
- What factors, beyond recency of last cue, influence the transiency of long-distance relationship? One possibility is that networks will have difficulty carrying information through parts of the script which have been poorly learned. For example, if the network never learns which *thing1* item to activate, then the *thing1* slot is a position of high error. Carrying information about gender through this high-error region could be problematic. Performance might be improved by changing the training regimen. Early in training, the network could be trained on sentences containing only dependencies between adjacent items. As these are mastered, non-adjacent dependencies could be added to the training set.

Viable connectionist models of natural language will require an understanding of the principles governing regularities *between* two representational systems as well as *within* a single representational system. Our long-term goal is to construct a model which maps between two systems. The contribution of the current work is that it explores some of the properties of a computational system that may eventually be a useful tool for achieving this goal.

REFERENCES

Elman, J. (1988). Finding structure in time. CRL Technical report 8801, Center for Research in Language, University of California, San Diego.

Elman, J. (1989). Structured representations and connectionist models. *Proceedings of the 11th Annual Cognitive Science Society Conference*. Hillsdale, New Jersey: Lawrence Erlbaum.

Norman, D. A. (1986). Reflections on cognition and parallel distributed processing. In J. L. McClelland and D. E. Rumelhart (eds.), *Parallel Distributed Processing, Vol. II*. Cambridge, MA: MIT Press.

Servan-Schreiber, D., Cleeremans, A., & McClelland, J.L. (1988). Encoding sequential structure in simple recurrent networks. Technical report CMU-CS-88-183, Carnegie Mellon University.