A Single Layer Network Model of Sentential Recursive Patterns

Lei Ding (dinglei@cse.ohio-state.edu)
Department of Computer Science & Engineering, The Ohio State University, 2015 Neil Ave, Columbus, OH 43210, USA

Simon Dennis (simon.dennis@gmail.com)
Department of Psychology, The Ohio State University, 1835 Neil Ave, Columbus, OH 43210, USA

Dennis N. Mehay (mehay@ling.ohio-state.edu)
Department of Linguistics, The Ohio State University, 1712 Neil Ave, Columbus, OH 43210, USA

Abstract

Recurrent connectionist models, such as the simple recurrent network (SRN, Elman, 1991), have been shown to be able to account for people's ability to process sentences with center embedded structures of limited depth without recourse to a competence grammar that allows unbounded recursion (Christiansen & Chater, 1999). To better understand the connectionist approaches to recursive structures, we analyze the performance of a single layer network architecture that employs decaying lexical context representation on three kinds of recursive structures (i.e., right branching, cross serial and center embedding). We show that the model with one input bank can capture one and two levels of right branching recursion, one level of center embedded recursion, but not two levels and cannot capture a single level of cross serial recursion. If one adds a second bank of input units with a different decay rate, the model can capture one and two levels of both center embedded and cross serial recursion. Furthermore, with this model the interclass difference of doubly cross serial patterns is greater than it is for center embedded recursion, which may explain why people rate these patterns as easier to process (Bach, Brown, & Marslen-Wilson, 1986).

Keywords: sentence processing; network model; decaying lexical contexts; linear separability.

Introduction

Chomsky (1957) argued that the presence of recursive structures such as center embedded clauses rules out associative explanations of the language processing mechanism. This argument has been challenged in many ways both by disputing the empirical claim that humans are capable of processing recursive structures (Reich, 1969) and the computational claim that associative mechanisms, particularly associative mechanisms that employ hidden unit representations in the connectionist tradition, are unable to process recursive structures, at least of the depth observed in human performance (Christiansen & Chater, 1999).

Early attempts to investigate the capabilities of connectionist networks to capture linguistic structure fell into two approaches (Christiansen & Chater, 1999). In the first approach, networks were provided with tagged datasets that provided information about the extent and identity of constituents (Chalmers, 1990; Pollack, 1988) and were required to generalize these mappings. While these models demonstrated the representational abilities of networks, the fact that they required labeled training data of a kind that is unlikely to be available to human learners meant that their relevance to the question of how linguistic structure is acquired was limited. A second approach involved learning simplified tasks such as identifying the $a^n b^n$ language from raw input strings using small networks (Wiles & Elman, 1995). This work demonstrated that recursive generalization was possible to a significant degree, but it remained unclear whether these results would apply to other kinds of recursion and with larger vocabularies.

Christiansen and Chater (1999) expanded previous work significantly by demonstrating that the simple recurrent network (Elman, 1991) was capable of capturing the three main kinds of recursive structures that were proposed by Chomsky (1957) as problematic for finite state systems. These were *counting* recursion (e.g. *ab*, *aabb*, *aaabbb*) of the kind studied by Wiles and Elman (1995); *identity* recursion (e.g. *abbabb*, *aabbaabb*) which captures cross serial structures found in Swiss German and in Dutch and *mirror* recursion (e.g. *abba*, *aabbaa*, *abbbba*) which can be interpreted as center embedding. In addition, Christiansen and Chater (1999) showed that the SRN predicted that cross serial structures should be easier to process than center embedded structures as has been demonstrated by Bach et al. (1986).

While the SRN provides a compelling proof of the capabilities of connectionist networks, its structure is not, in general, easy to analyze formally. Consequently, one must rely upon simulation results and it is not feasible to sample parameters such as initial weight vectors comprehensively. In this paper, we propose a single layer architecture that employs decaying input activations and analyze the linear separability and interclass distance of patterns in each of the recursion classes. We take linear separability as an indication of which patterns the model predicts should be able to be processed and the interclass distance as an indication of the ease with which that processing might occur. We start by outlining the model and then explore its performance at each level of recursion.

A Single-layer Network Model

Our network model, as seen in Figure 1, is a softmax single layer neural network (Bridle, 1990). For each word w in a sentence, an input $\mathbf{x}$ fed to the network has a nonzero value at the i^{th} position, only when a word token of the i^{th} type in a given vocabulary V appears to its left. The strength of context word's input is decided by the probability density function of an exponential distribution taking the distance between the words as the argument. The exponential function we use takes into account all the words occurring in the left context of a

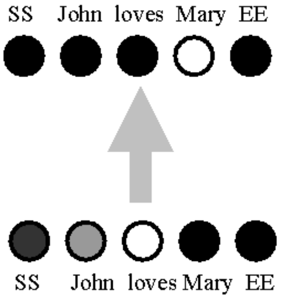

Figure 1: A schematic view of our network architecture with a single bank described in text. The left lexical context shown here is centered at "Mary". "SS" and "EE" refer to the start and the end of a sentence respectively.

word while assigning more importance to those closer to it.

Specifically, let a sentence be $w_1 w_2 w_3 \cdots w_i \cdots w_{n-1} w_n$, where w_i is a word and n is the total length. Then the left lexical context vector $\mathbf{x}$ for w_i is the following $|V|$-dimensional vector, where $|V|$ is the vocabulary size, and the j^{th} word in the vocabulary is V_j:

$$\sum_{j=1}^{|V|} \sum_{\substack{k=1 \\ w_k=V_j}}^{i-1} \gamma e^{-\gamma(i-k)} \mathbf{b}_j, \tag{1}$$

in which $\mathbf{b}_j$ is a unit basis vector in the j^{th} dimension, which has 1 in the j^{th} dimension and 0 elsewhere.

Outputs of the network are produced by multiplying the input vector $\mathbf{x}$ (of $|V|$ dimensions) with the learned weight matrix W (of size $|V| \times K$), and applying a multinomial logit function (Hastie, Tibshirani, & Friedman, 2001), which renders the output $\mathbf{y}$ a probability distribution over the K word types:

$$\mathbf{y}_i = \frac{e^{\mathbf{x}\cdot W_i}}{\sum_{j=1}^{K} e^{\mathbf{x}\cdot W_j}}, \tag{2}$$

where W_i is the i^{th} column of the matrix W. An important advantage of restricting our attention to a single layer network is that it allows us to analyze the learnability of different grammatical patterns in terms of the linear separability of the input patterns.

Sentential Recursive Patterns

We systematically study the left lexical context vectors derived from three common types of recursive structures found in human languages (where N_p and N_s stand for a plural and a singular noun respectively, and V_p and V_s stand for a plural and a singular verb respectively), namely:

1. *Right-branching* recursion:
   e.g., $N_p V_p N_s V_s$ girls like the boy that runs
2. *Cross-serial* recursion (derived from *identity* recursion):
   e.g., $N_s N_p V_s V_p$ the boy girls runs like
3. *Center-embedding* recursion (derived from *mirror* recursion)
   e.g., $N_s N_p V_p V_s$ the boy girls like runs

Therefore, $|V| = 4$ and $K = 2$ (only verbs are predicted). As an example of calculating lexical contexts, for the following center embedded sentence

$$\overbrace{\textit{The sea}}^{N_s}\ \overbrace{\textit{the mountains}}^{N_p}\ \overbrace{\textit{overlook}}^{V_p}\ \overbrace{\textbf{is}}^{V_s}\ \textit{blue},$$

the left lexical context vector centered at "is" for N_s, N_p, V_s and V_p is: $[\gamma e^{-3\gamma}, \gamma e^{-2\gamma}, 0, \gamma e^{-\gamma}]$.

The rest of this paper is organized as follows: We introduce the pattern subspaces occupied by certain context vectors and their dimensionality in the next section, followed by a formal characterization of learnability. We then present our computational results on the learnability and relative hardness of a single level of recursion ($N_s N_s V_s V_s$), and of two levels of recursion ($N_s N_s N_s V_s V_s V_s$). We conclude by summarizing the results and comparing them to human performance.

Pattern Subspaces and Dimensionality

For a single level of recursion, center embedding patterns are:

$$N_s N_s V_s V_s,\ \ N_s N_p V_p V_s,$$
$$N_p N_s V_s V_p,\ \ N_p N_p V_p V_p.$$

Noting that only the verbs are predictable, we generate the exponential distribution weighted left lexical context vectors at both the verbs for each pattern, leading to a total of 8 context vectors. The same is true for cross serial and right branching patterns with one level of recursion. For double center embedding, the patterns are:

$$N_s N_s N_s V_s V_s V_s,\ \ N_s N_s N_p V_p V_s V_s,$$
$$N_s N_p N_s V_s V_p V_s,\ \ N_s N_p N_p V_p V_p V_s,$$
$$N_p N_s N_s V_s V_s V_p,\ \ N_p N_s N_p V_p V_s V_p,$$
$$N_p N_p N_s V_s V_p V_p,\ \ N_p N_p N_p V_p V_p V_p.$$

We generate the left lexical context vectors at all the verbs for each pattern, leading to a total of 24 vectors. Similarly, we generate 24 vectors for the cross serial and right branching cases. We use D_i to refer to the set of vectors for $i \in \{\text{r, s, c}\}$ denoting double right branching, cross serial and center embedding respectively (e.g. D_c refers to the set of 24 lexical context vectors from double center embedding).

Despite the fact that all the members in D_r, D_s and D_c are 4 dimensional, an analysis with singular value decomposition (SVD, Hastie et al., 2001) reveals a lower intrinsic dimensionality. Table 1 summarizes our findings. Similar results on dimensionality are observed for the patterns of a single level of recursion. Specifically, the set of right branching vectors lie on a three-dimensional subspace (hyperplane), as do cross

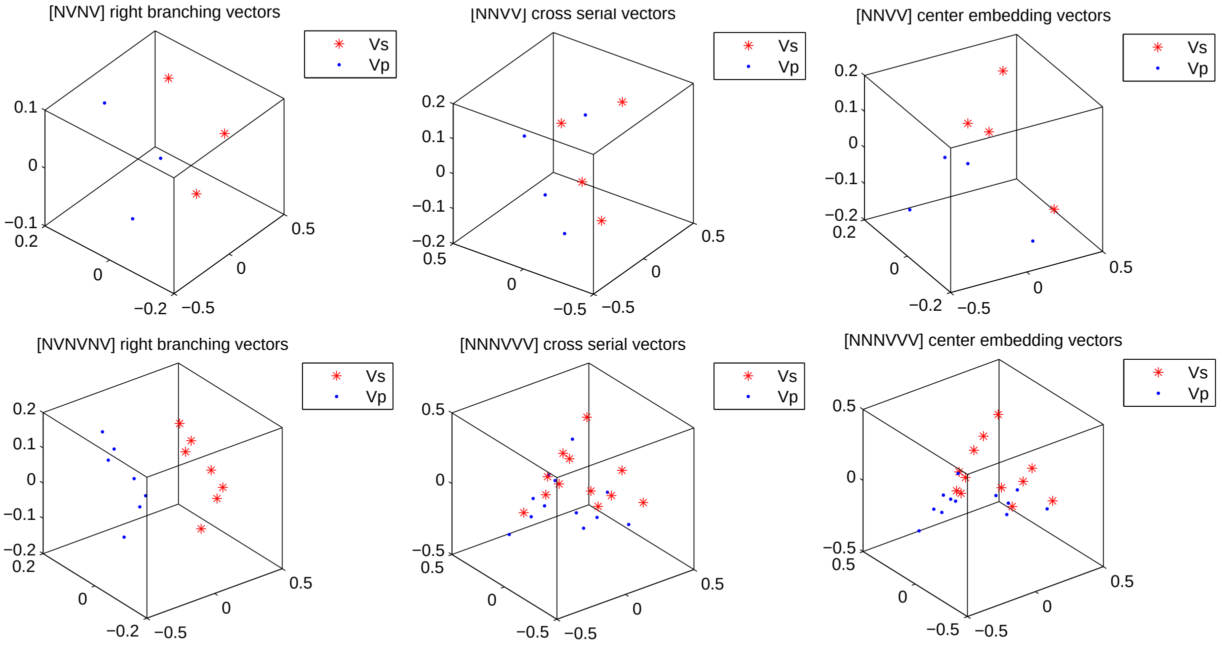

Figure 2: The three dimensional representations of the lexical context vectors from one level (top) and two levels (bottom) of recursion with the decay parameter being 0.5.

Set	Card.	Dim.
D_r	24	3
D_s	24	3
D_c	24	3
$D_r \cup D_s$	48	4
$D_r \cup D_c$	48	4
$D_s \cup D_c$ $(\gamma_s = \gamma_c)$	48	3
$D_s \cup D_c$ $(\gamma_s \neq \gamma_c)$	48	4
$D_r \cup D_s \cup D_c$	72	4

Table 1: Cardinality and dimensionality of context vectors. γ_s and γ_c are the decay parameters for D_s and D_c respectively.

serial vectors and center embedding vectors (see Figure 2). Thus, when we look at the linear separability of those lexical context vectors, it is equivalent to examine visually their 3D subspaces. Collectively, cross serial and center embedding vectors with the same decay parameter lie on the same hyperplane. However, the right branching vectors exist on a different hyperplane.

Characterizing Learnability

Our setting for characterizing learnability is that we have two classes of vectors D_1 and D_2 (in our case left lexical context vectors of V_s and V_p). We want to see if we can find a hyperplane that separates points in the two sets, and when they are separable, the minimum interclass distance indicates how far apart the two classes are in the space and also might indicate how easy it is to find a hyperplane that separates the classes. In addition, for the inseparable cases, we present an approach to identify the patterns that break the separability.

Linear Separability. A survey of linear separability tests is provided by Elizondo (2006). We use a simple method: Let D_1 and D_2 be the matrices of row vectors for the two classes respectively, and $\mathbf{1}$ be a vector of ones. A is defined as:

$$A = \begin{pmatrix} D_1 & \mathbf{1} \\ -D_2 & -\mathbf{1} \end{pmatrix}.$$

Linear separability is equivalent to $\{\mathbf{x} | A\mathbf{x} > \mathbf{0}\}$ is nonempty. Let $\mathbf{p} = (1, 1, \cdots, 1)^T$, $\alpha \in R$. A linear program for testing linear separability is:

$$\begin{aligned} &\min_{\mathbf{x},\alpha} \quad \alpha \\ &\text{subject to} \quad A\mathbf{x} + \alpha\mathbf{p} \geq \mathbf{1},\ \alpha \geq 0. \end{aligned} \tag{3}$$

Then the original sets D_1 and D_2 are linearly separable $\Longleftrightarrow$ The optimal $\alpha^* = 0$. The linear program is efficiently solved by using the Simplex method (Dantzig, 1963).

Minimum Interclass Distance. The minimum distance between two classes of vectors is defined as:

$$d_m = \min_{\mathbf{x} \in D_1, \mathbf{y} \in D_2} dist(\mathbf{x}, \mathbf{y}), \tag{4}$$

which characterizes how far the two classes are apart given that they are linearly separable.

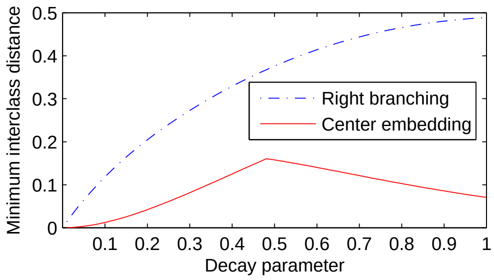

Figure 3: The minimum interclass distance with respect to the decay parameter for a single level of recursion (4 words).

Which Patterns Break the Separability? To gain additional insight into the model we can examine which patterns are responsible for preventing the pattern sets from being linearly separable. We invoke linear Support Vector Machines (SVMs, Cortes & Vapnik, 1995), which are now a set of well-studied techniques for regression and classification in the field of machine learning. The simplest form of SVMs for separating context vectors is:

$$\min \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i$$
$$\text{subject to} \quad \mathbf{w}\cdot\mathbf{x}_i - \mathbf{b} \geq 1 - \xi_i, i \in V_p$$
$$\mathbf{w}\cdot\mathbf{x}_i - \mathbf{b} \leq -1 + \xi_i, i \in V_s$$
$$\xi_i \geq 0,$$

where $\mathbf{x}_i$'s are the lexical context vectors, $\mathbf{w}$ is the linear weight vector, $\mathbf{b}$ is the bias and ξ_i's are the slack variables for the inseparable case. This SVM strives for a separating hyperplane with a reasonably large $C = 10^7$ for instance. If there exists no hyperplane that can split the examples from two classes, it will choose a hyperplane that splits the examples as cleanly as possible, while still maximizing the distance between the nearest cleanly split examples. Therefore, to find out which patterns break the separability, we can run the SVM on the context vectors and identify those vectors that the SVM makes a mistake on.

Results on Recursive Patterns

As plotted in Figure 2, for one level of recursion the right branching and center embedding vectors are linearly separable, while cross serial vectors are not, for decay parameters between 0.01 to 1. The change of the minimum interclass distance of right branching and center embedding vectors with respect to the decay parameter is plotted in Figure 3. For two levels of recursion, right branching vectors are separable, while the other two kinds of vectors are not (see Figure 2).

Clearly, the pattern of results differs from the empirical data in important ways. In particular, the cross serial patterns are not separable even for a single level of recursion. Consequently, we augmented the model by adding another set of input units, which operated in the same fashion as the first set except that the decay rate was varied indepedently.

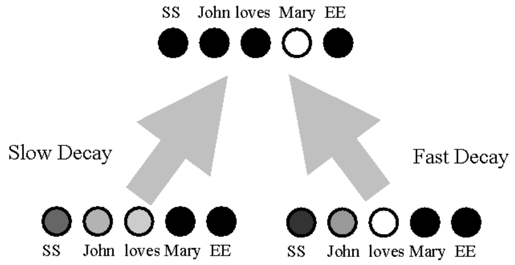

Figure 4: A schematic view of our model with two banks for inputs. The left lexical context shown is centered at "Mary".

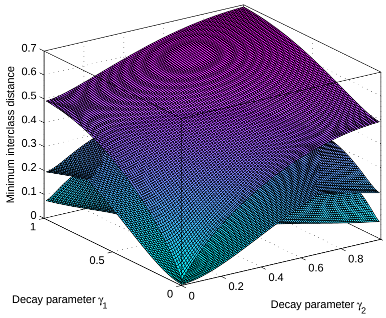

Figure 5: The minimum interclass distance with respect to the decay parameters. From top down, the surfaces are for right branching, cross serial and center embedding respectively.

Two Banks of Inputs

Unlike in slot-based networks, where the inputs from one set of units are transferred into adjacent slots as time progresses, in our model the two banks operate independently (see Figure 4). Representations are never transferred between the banks.

With two input banks, the outputs of the network are produced by multiplying the input vector (of $2|V|$ dimensions), which is the concatenation of $\mathbf{x}_1$ (with a decay parameter γ_1) and $\mathbf{x}_2$ (with a decay parameter γ_2), with the learned weight matrix W (of size $2|V| \times K$), and applying a multinomial logit function on the products, which does the following:

$$\mathbf{y}_i = \frac{e^{[\mathbf{x}_1,\mathbf{x}_2]\cdot W_i}}{\sum_{j=1}^{K} e^{[\mathbf{x}_1,\mathbf{x}_2]\cdot W_j}}, \qquad (5)$$

where $[\mathbf{x}_1, \mathbf{x}_2]$ represents the vector formed by concatenating $\mathbf{x}_1$ and $\mathbf{x}_2$, and W_i is the i^{th} column of the matrix W.

For a single level of recursion, we find that when using two banks of inputs with the decay parameters γ_1 and γ_2, cross serial vectors are linearly separable except when $\gamma_1 = \gamma_2$, and right branching and center embedding vectors are always separable, for $0.01 \leq \gamma_1 \leq 1$ and $0.01 \leq \gamma_2 \leq 1$. We then turn to the comparison of their minimum interclass distances. As plotted in Figure 5, the minimum interclass distance of right branching vectors is larger than that of cross serial vectors, which is larger than or equal to that of center embedding vectors, given the same decay parameters. Our results suggest

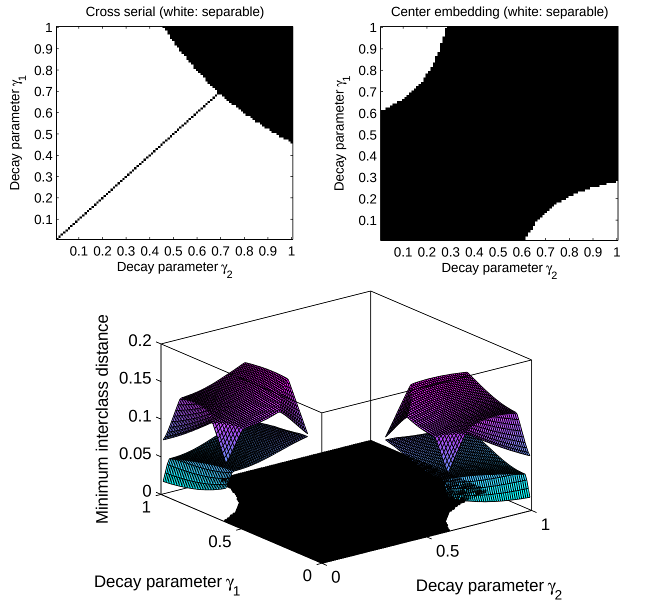

Figure 6: Top: linear separability as a function of decay parameters (white: separable). Bottom: the minimum interclass distance of cross serial and center embedding vectors in the parameter region (shown as white) where both of them are linearly separable. The top surface segments whose heights show the minimum interclass distance are those of cross serial and the bottom ones are those of center embedding.

that center embedding is the hardest form of recursive structure among the three being studied.

The results on two levels of recursion are plotted in Figure 6, which shows the relative hardness of cross serial and center embedding both from linear separability and from the minimum interclass distance when both of them are linearly separable. In the figure, the separable parameter regions are colored white, and the inseparable ones black. We find that in the whole parameter space $(0.01 \le \gamma_1 \le 1, 0.01 \le \gamma_2 \le 1)$, the minimum interclass distance of cross serial vectors is always larger than or equal to that of center embedding vectors given the same (γ_1, γ_2) pair. On the other hand, right branching vectors are always separable. Again, we can see that within this parameter region, center embedding is the hardest form of structure among the three being examined.

Patterns that Break the Separability. Using the aforementioned SVMs, we are able to identify the patterns that break the separability for two-level cross serial and center embedding structures. We define the breakers for linear separability to be the *minimal* set of vectors whose removal renders the previously inseparable set linearly separable, which are the same as those misclassified vectors reported by the SVM running at the decay parameters right next to the separable region (see Figure 6).

For two-level cross serial, the breakers are (where bold face symbols denote the location at which a context vector is generated):

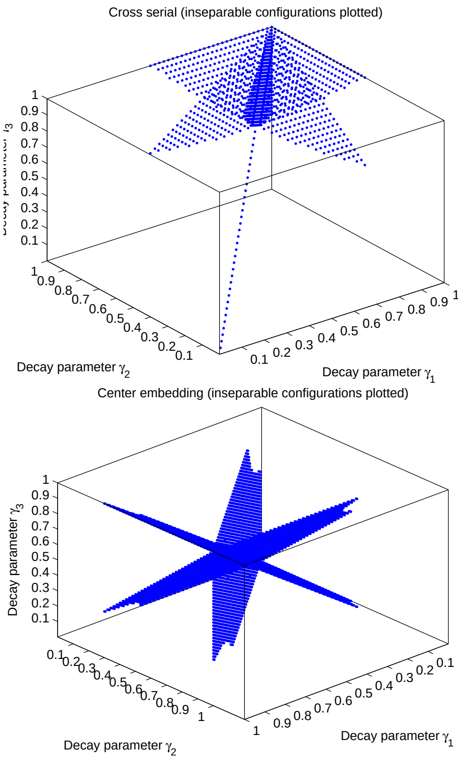

Figure 7: The inseparable configurations with three banks of inputs for two levels of recursion plotted as blue points. Top: cross serial patterns; Bottom: center embedding patterns. Down-sampling of the whole set of configurations is used for better visualization.

$$a = N_pN_sN_p\mathbf{V_p}V_sV_p, \quad b = N_sN_pN_s\mathbf{V_s}V_pV_s.$$

The removal of the above vectors renders the set of context vectors separable, for the two decay parameters ranging from 0.01 to 1, except when the two parameters are identical.

For two-level center embedding, the breaker vectors are:

$$a = N_sN_pN_sV_s\mathbf{V_p}V_s, \quad b = N_pN_sN_pV_p\mathbf{V_s}V_p,$$
$$c = N_sN_sN_p\mathbf{V_p}V_sV_s, \quad d = N_pN_pN_s\mathbf{V_s}V_pV_p.$$

Similarly, the removal of the above vectors renders the set separable, for the same parameter region as above.

Three Banks of Inputs

We also study the linear separability of patterns with three banks of inputs. Similar to Eqn. (5), the input vector here is the concatenation of three individual context vectors with γ_1, γ_2 and γ_3 as the decay parameters respectively.

Our results on two levels of recursion, as shown in Figure 7, suggest that the inseparable configurations are all caused by the degenerate cases, where at least two of the decay parameters are the same. Other than those cases, three banks of inputs are systematically able to separate the cross serial and center embedding patterns with two levels of recursion. Two-level right branching patterns are always separable with three banks.

Conclusions

We have presented a single layer network architecture to account for people's ability to understand recursive structures in language. Rather than posit separate slots or employing a recurrent network, we propose that sequence information is retained using a simple decay mechanism (c.f. ordinal models of serial order such as the primacy model (Page & Norris, 1998)). One might imagine that such a mechanism would ensue as the firing rates of populations of neurons decreased as a function of the time since the corresponding word was presented.

By restricting ourselves to the single layer case, we have been able to provide a more precise analysis than would otherwise be possible. The main separability results are summarized in Table 2.

# of levels (c.s.)	1-bank	2-bank	3-bank
1	never	always	always
2	never	depends	always

# of levels (c.e.)	1-bank	2-bank	3-bank
1	always	always	always
2	never	depends	always

Table 2: Separability of cross serial (top) and center embedding (bottom) patterns. "Always" means true for all the nondegenerative parameters tested, "never" means false for all the parameters tested, and "depends" means depending on the decay parameters used.

The decaying representation has a number of useful properties for accounting for human performance with recursive patterns. Specifically, with two banks of inputs, for both one and two levels of recursion, our model can rank the three kinds of recursion in an increasing order of hardness as: right branching, cross serial and center embedding, which is consistent with the results established by Bach et al. (1986).

References

Bach, E., Brown, C., & Marslen-Wilson, W. (1986). Crossed and nested dependencies in German and Dutch: A psycholinguistic study. *Language and Cognitive Processes*, *1*(4), 249–262.

Bridle, J. S. (1990). Probabilistic interpretation of feedforward classification network outputs with relationships to statistical pattern recognition. In F. F. Soulie & J. Herault (Eds.), *Neurocomputing: Algorithms, architectures and applications* (p. 227-236). Springer-Verlag.

Chalmers, D. J. (1990). Syntactic transformations on distributed representations. *Connection Science*, *2*, 53–62.

Chomsky, N. (1957). *Syntactic structures*. Mouton: The Hague.

Christiansen, M. H., & Chater, N. (1999). Towards a connectionist model of recursion in human lingusitic performance. *Cognitive Science*, *23*, 157–206.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Dantzig, G. B. (1963). *Linear programming and extensions*. Princeton, NJ: Princeton University Press.

Elizondo, D. A. (2006). The linear separability problem: Some testing methods. *IEEE Transactions on Neural Networks*, *17*(2), 330–344.

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

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning*. New York: Springer.

Page, M., & Norris, D. (1998). The primacy model: A new model of immediate serial recall. *Psychological Review*, *105*(4), 761–781.

Pollack, J. B. (1988). Recursive auto-associative memory: Devising compositional distributed representations. In *Proceedings of annual meeting of the cognitive science society*.

Reich, P. (1969). The finiteness of natural language. *Language*, *45*, 831–843.

Wiles, J., & Elman, J. L. (1995). Learning to count without a counter: A case study of synamics and activation landcapes in recurrent networks. In *Proceedings of annual meeting of the cognitive science society*.