Fractal Unfolding: A Metamorphic Approach to Learning to Parse Recursive Structure*

Whitney Tabor whitney.tabor@uconn.edu
Pyeong Whan Cho pyeong.cho@uconn.edu
Emily Szkudlarek emilyszkudlarek@gmail.com

Department of Psychology and Cognitive Science Program
University of Connecticut
406 Babbidge Road
Storrs, CT 06269-1020

Abstract

We describe a computational framework for language learning and parsing in which dynamical systems navigate on fractal sets. We explore the predictions of the framework in an artificial grammar task in which humans and recurrent neural networks are trained on a language with recursive structure. The results provide evidence for the claim of the dynamical systems models that grammatical systems continuously metamorphose during learning. The present perspective permits structural comparison between the recursive representations in symbolic and neural network models.

1 Introduction

Some loci in the phrase structure systems of natural languages appear to employ center embedding recursion (Chomsky, 1957), or at least an approximation of it (Christiansen and Chater, 1999). For example, one can embed a clause within a clause in English, using the object-extracted relative clause construction (e.g., *the dog that the goat chased barked.*). But such recursion does not appear in every phrase and may not appear in every language (Everett, 2005). Therefore, the system that learns natural languages must have a way of recognizing recursion when it occurs. We are interested in the problem, How does a language learner, seeing only a finite amount of data, decide on an unbounded recursive interpretation?

Here, we use the term "finite state" to refer to a system that can only be in a finite number of states. We use the term "recursion" to refer to situations in which multiple embeddings require the use of an unbounded symbol memory to keep track of unfinished dependencies.[1] We focus here on the case of center-embedding recursion, which can be generated by a context free grammar (one symbol on the left of each rule, finitely many symbols on the right) or a push-down automaton (stack memory + finite state controller) but not by a finite state device (Hopcroft and Ullman, 1979).

One natural approach to the recursion recognition problem, recently explored by Perfors et al. (2011), involves Bayesian grammar selection. Perfors et al.'s model considered a range of grammars, including both finite state and context free grammars. Their system, parameterized by data from English-speaking children in the Childes Database selected a context free grammar. Several features of this approach are notable: (i) There is a rich set of structural assumptions (the grammars in the pool of candidates). (ii) Because many plausible grammars generate overlapping data sets, a complexity ranking is also assumed and the system operates under Occam's Razor: prefer simpler grammars. (iii) Grammar selection and on-line parsing are treated as sep-

*This material is based on work supported by the National Science Foundation Grant No. 1059662. We thank the members of SOLAB who helped run the experiment: Olivia Harold, Milod Kazerounian, Emily Pakstis, Bo Powers, Kevin Semataska.

[1] This is a narrow construal of the term "recursion". Sometimes the term is used for any situation in which a rule can be applied arbitrarily many times in the generation of a single sentence, including finite-state cases.

In: R. Levy & D. Reitter (Eds.), *Proceedings of the 3rd Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2012)*, pages 41–50, Montréal, Canada, June 7, 2012.

arate problems in that the system is evaluated for coverage of the observed sentences, but the particular method of parsing plays no role in the selection process.

Here, we focus on a contrasting approach: recurrent neural network models discover the structure of grammatical systems by sequentially processing the corpus data, attempting to predict after each word, what word will come next (Elman, 1990; Elman, 1991). With respect to the properties mentioned above, the neural network approach has some advantages: (i) Formal analyses of some of the networks and related systems (Moore, 1998; Siegelmann, 1999; Tabor, 2009b) indicate that these models make even richer structural assumptions than the Bayesian approach: if the networks have infinite precision, then some of them recognize all string languages, including non-computable ones. For a long while, theorists of cognition have adopted the view that positing a restrictive hypothesis space is desirable—otherwise a theory of structure would seem to have little substance. However, if one offers a hypothesis about the organization of the hypothesis space, and a principle that specifies the way a learning organism navigates in the space, then the theory can still make strong, testable predictions. We suggest that assuming a very general function class is preferable to presupposing arbitrary grammar or class restrictions. (ii) The recurrent networks do not employ an independently defined complexity metric. Instead, the learning process successively breaks symmetries in the initially unbiased weight set, driven by asymmetries in the data. The result is a bias toward simplicity. We see this as an advantage in that the simplicity preference stems from the form of the architecture and learning mechanism. (iii) Word-by-word parsing and grammar selection occur as part of a single process—the network updates its weights every time it processes a word and this results in the formation of a parsing system. We see this as an advantage in that the moment-to-moment interaction of the system with data resembles the circumstances of a learning child.

On the other hand, there has long been a serious difficulty with the network approach: the network dynamics and solutions have been very opaque to analysis. Although the systems sometimes learn well and capture data effectively, they are not scientifically very revealing unless we can interpret them. The Bayesian grammar-selection approach is much stronger in this regard: the formal properties of the grammars employed are well understood and the selection process is well-grounded in statistical theory—e.g., Griffiths et al. (2010).

Here, we take advantage of recent formal results indicating how recurrent neural networks can encode abstract recursive structure (Moore, 1998; Pollack, 1987; Siegelmann, 1999; Tabor, 2000) An essential insight is that the network can use a spatial recursive structure, a fractal, to encode the temporal recursive structure of a symbol sequence. When the network is trained on short sentences exhibiting a few levels of embedding, it tends to generalize to higher levels of embedding, suggesting that it is not merely shaping itself to the training data, but discovers an abstract principle (Rodriguez et al., 1999; Rodriguez, 2001; Tabor, 2003; Wiles and Elman, 1995). During the course of learning, the fractal comes into being gradually in such a way that lower-order finite-state approximations to the recursion develop before higher-order structure does—a complexity cline phenomenon (Tabor, 2003).

We examined human and neural network learning of a recursive language with an artificial grammar paradigm, the Box Prediction paradigm. Whereas our previous investigations of this task (Cho et al., 2011) focused on counting recursion languages (only a single stack symbol is required to track the recursive dependencies), we provide evidence here for mirror recursion learning by a few participants (multiple stack symbols required). We show how the theory of fractal grammars can be used to hand wire a network that processes the recursive language of our task. We then provide evidence that a Simple Recurrent Network (Elman, 1990; Elman, 1991), trained on the same task, also develops a fractal encoding. Moreover, the network shows evidence of a embodying a complexity-cline—similarly complex grammars are adjacent in the parameter space. An individual differences analysis indicates that a similar pattern arises in the humans. We conclude that the network encodings can be formally related to symbolic recursive models, but are different in that learning occurs by continuous grammar metamorphosis.

2 The Box Prediction paradigm

Human participants sat in front of a computer screen on which five black outlines of boxes were displayed (Figure 1). When the participant clicked on the screen, one of the boxes changed color. The task was to indicate, by clicking on it, which box would change color next on each trial. The sequence of color changes corresponded to the structure of sentences generated by the center-embedding grammar in Table 1a. The sentences can be divided into embedding level classes. Level n sentences have (n-1) center-embedded clauses (Table 1b). There were three, distinct phases of the color-change sequence: during the first 60 trials, participants saw only Level 1 sentences. From trials 61 to 410, Level 2 sentences were introduced with increasing frequency. We refer to these two phases of presentation together as the "Training Phase". Starting at Trial 411, Level 3 sentences were included, along with more Level 1 and Level 2 sentences. We refer to the trials from 411 to 553, the end of the experiment, as the "Test Phase". Other than by their structural differences, these phases were not distinguished for the participants: the participants experienced them as one, long sequence of 553 trials. We introduced the deeper levels of embedding gradually because of evidence from the language acquisition literature (Newport, 1990), from the connectionist literature (Elman, 1993), and from the artificial grammar learning literature (Cho et al., 2011; Lai and Poletiek, 2011) that "starting small" facilitates learning of complex syntactic structures. Following standard terminology, we call the trials in which boxes 1 and 4 change colors "push" trials (because in a natural implementation of the grammar with a push-down automaton, the automaton pushes a symbol onto the stack at these trials). We call the trials in which boxes 2, 3, and 5 change color "pop" trials. The push trials were fairly unpredictable: the choice of whether to push 1 or 4 was approximately uniformly distributed throughout the experiment, and the choice about whether to embed was fairly random within the constraints of the "starting small" scheme described above. Because we did not want participants to have to guess at these nondeterministic events, we made the 1 and 4 boxes turn blue or green whenever they occurred and told the participants that they did not need to predict blue or green boxes. On the other hand, we wanted them to predict the pop trials whenever they occurred. Therefore, we colored boxes 2, 3, and 5 a shade of red whenever they occurred and told the participants that they should try to predict all boxes that turned a shade of red. When two of the same symbol occurred in a row (e.g., 1 1 2 2 5), we shifted the shade of the color of the repeated element so that participants would notice the change. To reinforce this visual feedback, a beep sounded on any trial in which a participant failed to predict a box that changed to a shade of red. Box 5 has a different structural status than the other boxes: it marks the ends of sentences. We included box 5, placing it in the center of the visual array, and making it smaller than the other boxes, to make the task easier relative to a pilot version in which Box 5 was absent.

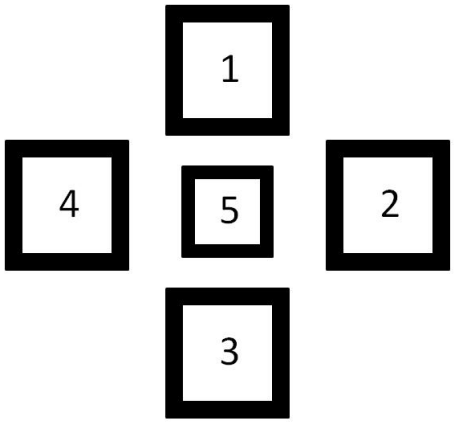

Figure 1: Structure of the display for the Box Prediction Task. The numerals were not present in the screen display shown to the participants.

2.1 Simulation Experiment

We employed Michal Cernansky's implementation of Elman (1990)'s Simple Recurrent Network (http://www2.fiit.stuba.sk/černans/main/download.html). The network had five input units, five output units and ten hidden units. Activations changed as specified in (1) and weights changed according to (2).

a.

Root	→	S 5
S	→	1 S 2
S	→	1 2
S	→	4 S 3
S	→	4 3

b.

Level 1	Level 2	Level 3
1 2 5	1 1 2 2 5	1 1 1 2 2 2 5
4 3 5	1 4 3 2 5	1 1 4 3 2 2 5
	4 1 2 3 5	1 4 1 2 3 2 5
	4 4 3 3 5	2 4 4 3 3 2 5
		...

Table 1: a. Grammar 1: a recursive grammar for generating the color change sequence employed in the experiment. "Root" is the initial node of every sentence generation process. *Null* stands for the empty string. b. Examples of Level 1, 2, and 3 sentences generated by Grammar 1.

$$\begin{aligned} \vec{h}(t) &= f(\mathbf{Whh} \cdot \vec{h}(t-1) + \mathbf{Whi} \cdot \vec{s}(t) + \vec{b}_h) \\ \vec{o}(t) &= f(\mathbf{Woh} \cdot \vec{h}(t) + \vec{b}_o) \\ f(x) &= \frac{1}{1+e^{-x}} \end{aligned} \tag{1}$$

$$\Delta w_{ij} \propto -\frac{\partial E}{\partial w_{ij}} \tag{2}$$

Here, $\vec{s}(t)$ is the vector of input unit activations at time step t, $\mathbf{Whi}$ are the weights from input to hidden units, $\mathbf{Whh}$ are the recurrent hidden connections, and $\mathbf{Woh}$ connect hidden to output.

On each trial, the input to the network was an indexical bit vector corresponding to one of the five sentence symbols. The task of the network was to predict, on its output layer, what symbol would occur next at each point. The sequence of symbols was modeled on the sequence presented to the human participants as follows: the human sequence was divided into 14 nearly equal-length segments, each with a whole number of sentences (the first 11 segments corresponded to the Training Phase and the last 3 to the Test Phase). Each segment contained approximately ten sentences. For each segment, 400 sentences were sampled randomly according to the distribution of types found in the segment. These groups of 400 were concatenated end to end to form the training sequence for the network (a total of 22398 trials).

The error gradient of equation (2) was approximated using Backpropagation Through Time (Rumelhart et al., 1986) with eight time steps unfolded. To simulate the absence of negative feedback on push trials in the human experiment, the network error signal on push trials was set to zero. The constant of proportionality in equation 2 (the "learning rate") was set to 0.4.

3 Fractal Encoding of Recursive Structure in Neural Ensembles

In the past several decades, a number of researchers (Moore, 1998; Pollack, 1987; Siegelmann, 1996; Siegelmann and Sontag, 1994; Tabor, 2000) have developed devices for symbol processing which compute on finite-dimensional complete metric spaces (distance is defined, no points are "missing"— (Bryant, 1985)), like the neural networks considered here. A common strategy in all of these proposals is the use of spatially recursive sets—i.e., fractals—to encode the temporal recursive structure in symbol sequences. For example, Tabor (2000) defines a *Dynamical Automaton* (or DA), M, as in (3).

$$M = (H, F, P, \Sigma, IM, x_0, FR) \tag{3}$$

Here, H is a complete metric space (Bryant, 1985; Barnsley, 1993). F is a finite list of functions $f_i : H \rightarrow H$, P is a partition of the metric space, Σ is a finite symbol alphabet, IM is an *Input Map*—that is, a function from symbols in Σ and compartments in P to functions in F. The input to the machine is a finite string of symbols. The machine starts at x_0 and invokes functions corresponding the symbols in the input in the order in which they occur. If, when the last symbol has been presented, the system is in the region $FR \subseteq H$, then the DA *accepts* the string.

Table 3 specifies DA 1, a dynamical automaton that recognizes (and generates) the language of Grammar 1. A good way of understanding the principle underlying this mechanism is to note that a pushdown automaton (PDA) (Hopcroft and Ullman, 1979) for processing this language must employ a stack alphabet with one symbol for tracking"1" and another for tracking "4". (See Table 3). If DA 1 is to successfully process the same language, it must distinguish at least the states that PDA 1 distinguishes

(PDA 1 is minimal in this sense). DA 1 does this by executing state transitions analogous to the push and pop operations of the PDA, arriving in its final region when the PDA is in an accepting state. Figure 3 shows the correspondence between machine states of PDA 1 and points in the metric space H that underlies DA 1's language recognition capability. This figure makes it clear that DA 1 is structurally equivalent to PDA 1.

The computing framework discussed here is very general. One can construct a fractal grammar that generates any context free language (Tabor, 2000). In fact, similar mechanisms recognize and generate not only all computable languages but all languages of strings drawn from a finite alphabet (Moore, 1998; Siegelmann, 1999; Siegelmann and Sontag, 1994). Wiles & Elman (1995) and Rodriguez (2001) showed that an SRN trained on a counting recursion language ($a^n b^n$) uses a fractal principle to keep track of the embeddings and generalizes to deeper levels of embedding than those found in its training set. (Tabor et al., 2003) showed that a gradient descent mechanism operating in the parameter space of a fractal grammar model discovered close approximations of several mirror recursion languages. These findings suggest that the fractal solutions are stable equilibria ("attractors") of recurrent network gradient descent learning processes (Tabor, 2011). This observation argues against a widespread belief about neural networks that they are blank slate architectures, only performing "associative processing" without structural generalization (Fodor and Pylyshyn, 1988). It suggests a close relationship between the classical theory of computation and neural network models even though the two frameworks are not equivalent (Siegelmann, 1999; Tabor, 2009a) .

The results of Tabor (2003) indicate that network learning proceeds along a complexity cline: sentences with lower levels of embedding are correctly processed before sentences with higher levels of embedding. This indicates that there are proximity relationships in the network parameter space: parameterizations that parse successively deeper levels of embedding are adjacent to each other. In the next section, we investigate the outcome of the SRN learning experiment with the Box Prediction training data, first testing for evidence that the network forms a fractal code, then testing for a proximity effect consistent with the complexity cline prediction.

$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$
$Q = \{q_1, q_2, q_3\}$
$\Sigma = \{1, 2, 3, 4, 5\}, \Gamma = \{B, O, F\}$
$q_0 = q_3$, $Z_0 = B$, $F = B$
$\delta(q3, 1, B) = (q1, OB)$, $\delta(q3, 4, B) = (q1, FB)$
$\delta(q1, 1, O) = (q1, OO)$, $\delta(q1, 4, O) = (q1, FO)$
$\delta(q1, 1, F) = (q1, OF)$, $\delta(q1, 4, F) = (q1, FF)$
$\delta(q1, 2, O) = (q2, \epsilon)$, $\delta(q2, 2, O) = (q2, \epsilon)$
$\delta(q1, 3, F) = (q2, \epsilon)$, $\delta(q2, 3, F) = (q2, \epsilon)$
$\delta(q2, 5, B) = (q3, B)$

Table 3: PDA 1. A Pushdown Automaton for processing the language of Grammar 1. "O" is pushed on "1". "F" is pushed on "4".

4 Results: Simple Recurrent Network Box Prediction

We trained 71 networks, corresponding to the 71 human participants on the sequence described above in Section 2. The networks all used the same architecture, but differed in the values of their random initial weights and the precise ordering of the training sentences (although all used the same progressive scheme described above). To approximate the observed variation in human performance, each network also had gaussian noise with constant variance added to the weights with each new word input. The variance values were sampled from the uniform distribution on [0,4]. This range was chosen to produce a mean (57%) and standard deviation (20%) similar to that of the humans at the end of training ($M = 51\%, SD = 21\%$).

Unlike some of the humans, none of the networks generalized immediately to Level 3 sentences on the first try. Nevertheless, several of them learned to parse the Level 3 sentences with very few errors by the end of the "Test Phase". To determine accuracy of a deterministic transition, we normalized the network output vector by dividing all the outputs by the sum of the outputs. If the highest normalized activation was on the correct transition, we counted the transition as accurate. When tested on all eight types of Level 3 sentences, the top 4 networks made 1, 3, 3, and 3 errors among the 56 transitions in this sen-

Compartment	Symbol	Function
$h_1 > 0 \;\&\; h_2 > 0$	1	$\vec{h} \leftarrow \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \vec{h} + \begin{pmatrix} 1 \\ 0 \end{pmatrix}$
$h_1 > 0 \;\&\; h_2 > 0$	4	$\vec{h} \leftarrow \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \vec{h} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
$h_1 > 1$	2	$\vec{h} \leftarrow \begin{bmatrix} -2 & 0 \\ 0 & -2 \end{bmatrix} \vec{h} + \begin{pmatrix} 2 \\ 0 \end{pmatrix}$
$0 < h_1 < 1$	3	$\vec{h} \leftarrow \begin{bmatrix} -2 & 0 \\ 0 & -2 \end{bmatrix} \vec{h} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
$h_1 < -1$	2	$\vec{h} \leftarrow \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \vec{h} + \begin{pmatrix} 2 \\ 0 \end{pmatrix}$
$-1 < h_1 \;\&\; h_1 < 1$	3	$\vec{h} \leftarrow \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \vec{h} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
$h_1 = -1 \;\&\; h_2 = -1$	5	$\vec{h} \leftarrow \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \vec{h} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

Table 2: Input Map for DA 1. The automaton starts at the point, (1, 1). It's Final Region is also (1, 1).

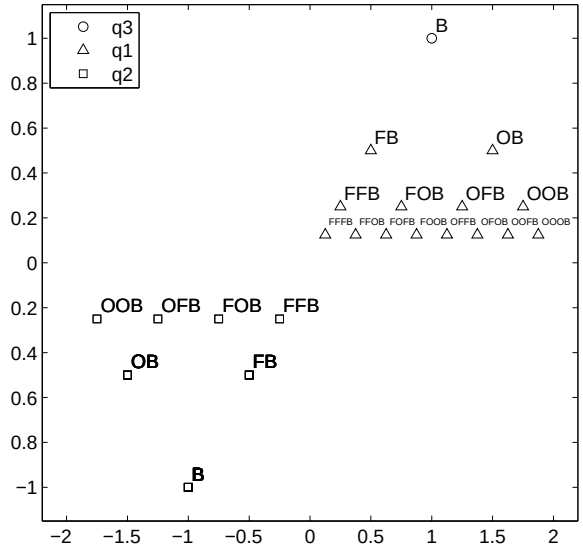

Figure 2: Correspondence between states of DA 1 and PDA stack states.

tence set.

We hypothesized that the networks were approximating a fractal grammar with the same qualitative structure as that of DA 1, possibly in more than two dimensions. We sought two kinds of evidence: "linear separability" and "branching structure". For "linear separability", we asked if the SRN states corresponding to a particular point in DA 1 (state of PDA 1) were clustered so as to be linearly separable from SRN states corresponding to a different point. Two sets A and B of points in a vector space of dimension n are *linearly separable* if there is an $n-1$ dimensional hyperplane in the space with all the points of A on one side of it and all the points of B on the other. In fractal grammar parsing, pairwise linear separability suffices to distinguish the machine states (Tabor, 2000). Among the cases where more than one sample point corresponded to the same PDA state, an average of 17.6/22 were pairwise linearly separable from the other groups. In six of the networks, all the multi-element clusters were pairwise linearly separable. This finding lends support to the claim that the networks approximate fractal grammars.

For "branching structure", we asked if the deployment of these (largely) linearly separable clusters corresponded to the branching structure of the fractal of DA 1. In particular, for each cluster corresponding to a DA 1/PDA 1 state with more than one symbol on the stack in PDA 1, we considered all the clusters with one-fewer symbols on the stack. We asked if the nearest cluster with one fewer symbols on the stack corresponded to the nearest one-fewer stack symbol point in DA 1. In Level 1 to and 3 sentences, there are 20 such states to consider. Across networks, the average rate of unexpected proxim-

ity relationships was 3.5/20 (SD = 1.7). The best networks we observed under this training method (noise reduced to 0) generated only 1 unexpected proximity relationship. These results also indicate a close correspondence between the organization of the network and fractal grammars.

Up to this point, the evidence we have been presenting has supported a formal correspondence between SRNs and fractal grammars. In the final part of this section, we consider one prediction of the network approach that is not obviously predicted by symbolic grammar mixture accounts like the Bayesian model discussed in the introduction.

Tabor (2003) shows how a fractal for processing another recursive language (similar to the language of Grammar 1) arises by gradual metamorphosis of (Stage I) a single point into (Stage II) a line of points, then into (Stage III) an infinite lattice, then into (Stage IV) a fractal with overlapping branches and finally into (Stage V) the fully-formed fractal that very closely captures the recursive embedding structure. During Stage IV, the system correctly processes shallow levels of embedding but fails to process deeper levels of embedding. As the metamorphosis progresses, this Fractal Learning Neural Network (FLNN) becomes able to process deeper and deeper levels at an accelerating rate such that, after finite time, it reaches a point where it is effectively processing all levels, indicating a continuous complexity cline in parameter space. An empirical implication is that a network that has mastered n levels of embedding, for n a natural number, will more easily (with less weight change) master n+1 levels of embedding than one that has mastered fewer than n.

To see if the SRN's complexity cline predictions are in line with those of the FLNNs, we correlated the network's performance at the end of the Training Phase with its performance in the Test Phase. For this purpose, we defined the training performance as the mean prediction accuracy across all predictable transitions of Level 1 and 2 sentences in the fourth quarter of the training phase. The Test Phase performance was defined in two different ways. It was defined as the mean accuracy across novel but predictable transitions (a) in all Level 3 sentences in the test phase or (b) only in the first instances of four different Level 3 sentences. We used the second measure because the networks and humans continue to learn in the Test Phase: correlation of training performance with measure (a) might stem from learning facility alone; correlation with (b) indicates generalization ability. Both tests showed significant correlation (a: $r(69) = 0.98, p < .0001$; b: $r(69) = 53, p < .0001$). These results are consistent with the claim that the SRN induces a complexity cline similar to that induced by the fractal learning networks..

To consider how well this prediction distinguishes the fractal learning framework from other approaches to grammar learning, we now consider the Bayesian grammar selection model of (Perfors et al., 2011). We consider this case, which is naturally related to our focus, as a first step toward developing concrete approaches within the Bayesian framework that could address the issues raised by the Box Prediction findings.

Perfors et al.'s model is also concerned with the induction of recursive grammatical systems from language data. They presented samples from the Childes Database (MacWhinney, 2000) to their model over 6 stages, where each stage sampled the corpus more thoroughly than the last. This sampling method generally caused each stage to have heavier sampling of deeper recursive structures than the previous stage because the deeper recursive structures are less frequent in the master corpus. The Bayesian model selects finite-state grammars during the earlier stages and then prefers recursive grammars during the later stages. This shift occurs because, as the sampling goes deeper, the finite state systems need to employ many additional productions to handle the burgeoning variety of collocations, while the recursive grammars can handle them with few rules, so the model's anti-complexity bias causes it to prefer the recursive grammars (Perfors et al., p. 320). It seems likely that a version of their model, applied to the training data in our experiment, would select finite-state grammars during the Training Phase and the switch to a recursive grammar in the Test Phase. Perfors et al. did not consider the question of individual differences. We can think of one way that the basic correlational finding reported for the SRNs would obtain in the Bayesian system (finding (a) above): if the perception of the stimuli by some models was noisier than that of others, then one ex-

pects the general correlation between Training and Test performance to obtain: the noise interferes similarly with both phases so correlated accuracy is observed. It is not as clear to us that the Bayesian system will predict finding (b), which shows that first-try performance on novel structures is better for people who show better Training performance.[2] There does not appear to be a proximity relationship between grammars in Perfors et al.'s model as there is in the network models. Thus, if it predicts this effect, then it would have to do so for a different reason, a point worthy of further research.

5 Results: Human Box Prediction

Seventy-one undergraduate students in the University of Connecticut participated in the experiment for course credit. The range of human performance was substantial. The mean correct performance on 37 predictable trials during the last 100 trials of training was 51% (SD = 21%). Despite this overall low rate of performance at test, there was a subset of people who learned the training grammar well by the end of training.

Twelve of the 71 participants, scored over 80% correct on the pop trials within the last 100 training trials. 80% is the level of correct performance that a particular finite-state device we refer to as the "Simple Markov Model" would yield during these 100 trials. The Simple Markov Model predicts 2 after 1, 3 after 2, 4 after 3, and 1 after 4. The two top scorers among these twelve generalized perfectly to each first instance of the four Level 3 types in the test phase. If, contrary to our hypothesis, all 12 were using finite state mechanisms, and they guessed randomly on novel transitions, the chances of observing 2 or more perfect scorers would be 0.9% (p = .009). We take this as evidence that the two strongest generalizers developed a representation closely approximating a recursive system.

Performance at the end of training correlated with accuracy on 24 novel transitions in Level 3 sentences at test ($r(69) = 0.72, p < .0001$). This corresponds to test (a) of the SRN Results section above, suggesting some kind of grammar proximity model. Regarding (b), accuracy on the 8 novel transitions in the 4 first instances of novel Level 3 sentences also correlated with the performance at the end of training, $r(69) = 0.57, p < .0001$. These results lend some empirical support to the complexity cline predictions of the fractal model.

6 General Discussion

We studied the learning of recursion by training Simple Recurrent Networks (SRNs) and humans in an artificial grammar task. We described metric space computing models that navigate on fractal sets and noted a complexity cline phenomenon in learning (learning of lower embeddings facilitates the learning of higher ones). Previous work in this area has focused on counting recursion languages. Here, we explored learning of a mirror recursion language. We showed that the SRN hidden unit representations had clustering and branching structure approximating the predictions of the fractal grammar model. They also showed evidence of the complexity cline. The human learning results on the same language provided evidence that at least a few people inferred a recursive principle for the mirror recursion language. The complexity cline prediction was also borne out by the human data: not only did performance on lower levels of embedding correlate with performance on higher levels of embedding, but it predicted generalization behavior, suggesting that the representation continuously metamorphoses from a finite-state system into an infinite state system.

We identified one closely related Bayesian grammar induction model (Perfors et al., 2011) which seems well positioned to make similar, but probably not the same, predictions about phenomenon of infinite state language learning. We suggest that further exploration of the relationship between the Bayesian models and the recurrent neural network models will be helpful. A novel claim of the present work is that they it is possible to compare recurrent neural network models and symbolic structure models on the same terms. We suggest that further examination of this relationship may be helpful in addressing the challenging problems of complex language learning.

[2] This is not single-trial learning. It is immediate generalization to unseen cases.

References

Michael Barnsley. 1993. *Fractals Everywhere, 2nd ed.* Academic Press, Boston.

Victor Bryant. 1985. *Metric Spaces. Iteration and Application*. Cambridge University Press, Cambridge, UK.

Pyeong Whan Cho, Emily Szkudlarek, Anuenue Kukona, and Whitney Tabor. 2011. An artificial grammar investigation into the mental encoding of syntactic structure. In Laura Carlson, Christoph Hoelscher, and Thomas F. Shipley, editors, *Proceedings of the 33rd Annual Meeting of the Cognitive Science Society (Cogsci2011)*, pages 1679–1684, Austin, TX. Cognitive Science Society. Available online at http://palm.mindmodeling.org/cogsci2011/.

Noam Chomsky. 1957. *Syntactic Structures*. Mouton and Co., The Hague.

Morten H. Christiansen and Nick Chater. 1999. Toward a connectionist model of recursion in human linguistic performance. *Cognitive Science*, 23:157–205.

Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science*, 14:179–211.

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

Jeffrey L. Elman. 1993. Learning and development in neural networks: the importance of starting small. *Cognition*, 48:71–99.

Daniel L. Everett. 2005. Cultural constraints on grammar and cognition in Piraha: another look at the design features of human language. *Current Anthropology*, 46(4):621–646, August.

J. A. Fodor and Z. W. Pylyshyn. 1988. Connectionistm and cognitive architecture: A critical analysis. *Cognition*, 28:3–71.

Thomas L. Griffiths, Nick Chater, Charles Kemp, Amy Perfors, and Joshua B. Tenenbaum. 2010. Probabilistic models of cognition: exploring representations and inductive biases. *Trends in Cognitive Sciences*, 14(8):357–364.

John E. Hopcroft and Jeffrey D. Ullman. 1979. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Menlo Park, California.

Jun Lai and Fenna H. Poletiek. 2011. The impact of adjacent-dependencies and staged-input on the learnability of center-embedded hierarchical structures. *Cognition*, 118(2):265–273, February.

B. MacWhinney. 2000. *The CHILDES project: Tools for analyzing talk*. Lawrence Erlbaum Associates, Mahwah, NJ, third edition.

Cris Moore. 1998. Dynamical recognizers: Real-time language recognition by analog computers. *Theoretical Computer Science*, 201:99–136.

Elissa L. Newport. 1990. Maturational constraints on language learning. *Cognitive Science*, 14(1):11–28, March.

Amy Perfors, Joshua B. Tenenbaum, and Terry Regier. 2011. The learnability of abstract syntactic principles. *Cognition*, 118(3):306–338, December.

Jordan Pollack. 1987. On connectionist models of natural language processing. Unpublished doctoral dissertation, University of Illinois.

Paul Rodriguez, Janet Wiles, and Jeffrey Elman. 1999. A recurrent neural network that learns to count. *Connection Science*, 11(1):5–40.

Paul Rodriguez. 2001. Simple recurrent networks learn context-free and context-sensitive languages by counting. *Neural Computation*, 13(9):2093–2118.

David E. Rumelhart, Geoffrey E. Hinton, and R. J. Williams. 1986. Learning internal representations by error propagation. In David E. Rumelhart, James L. McClelland, and the PDP Research Group, editors, *Parallel Distributed Processing, v. 1*, pages 318–362. MIT Press.

H. T. Siegelmann and E. D. Sontag. 1994. Analog computation via neural networks. *Theoretical Computer Science*, 131:331–360.

Hava Siegelmann. 1996. The simple dynamics of super Turing theories. *Theoretical Computer Science*, 168:461–472.

Hava T. Siegelmann. 1999. *Neural Networks and Analog Computation: Beyond the Turing Limit*. Birkhäuser, Boston.

Whitney Tabor, Bruno Galantucci, and Daniel Richardson. 2003. Evidence for self-organized sentence processing: Local coherence effects. Submitted manuscript, University of Connecticut, Department of Psychology: See http://www.sp.uconn.edu/ ps300vc/papers.html.

Whitney Tabor. 2000. Fractal encoding of context-free grammars in connectionist networks. *Expert Systems: The International Journal of Knowledge Engineering and Neural Networks*, 17(1):41–56.

Whitney Tabor. 2003. Learning exponential state growth languages by hill climbing. *IEEE Transactions on Neural Networks*, 14(2):444–446.

Whitney Tabor. 2009a. Affine dynamical automata. Ms., University of Connecticut Department of Psychology.

Whitney Tabor. 2009b. A dynamical systems perspective on the relationship between symbolic and non-symbolic computation. *Cognitive Neurodynamics*, 3(4):415–427.

Whitney Tabor. 2011. Recursion and recursion-like structure in ensembles of neural elements. In H. Sayama, A. Minai, D. Braha, and Y. Bar-Yam, editors, *Unifying Themes in Complex Sys-*

tems. Proceedings of the VIII International Conference on Complex Systems, pages 1494–1508, Cambridge, MA. New England Complex Systems Institute. http//necsi.edu/events/iccs2011/proceedings.html.

Janet Wiles and Jeff Elman. 1995. Landscapes in recurrent networks. In Johanna D. Moore and Jill Fain Lehman, editors, *Proceedings of the 17th Annual Cognitive Science Conference*. Lawrence Erlbaum Associates.