Towards Emergent Strong Systematicity in a Simple Dynamical Connectionist Network

Yuuya Sugita
Department of Psychology III (Cognitive Psychology)
University of Würzburg
Röntgenring 11, 97070 Würzburg, Germany
yuuya.sugita@gmail.com

Martin V. Butz
Department of Psychology III (Cognitive Psychology)
University of Würzburg
Röntgenring 11, 97070 Würzburg, Germany
butz@psychologie.uni-wuerzburg.de

Abstract

One of the most striking features of human language is its systematicity. This paper focuses on a fundamental, higher-order systematicity described by word classes. We investigate if such higher-order systematicity can be learned from symbol-based examples alone using a rather simple Jordan-type recurrent neural network (RNN). The network was allowed to keep a vector, which represents a suitable starting state for the production of a particular dynamic. With this representation, the network was able to reproduce particular sentences in a deterministic manner. We also show that the trained network can acquire representations of word classes to a certain extent. The analysis reveals that the higher-order grammatical constraints are realized by means of self-organized sub-symbolic dynamics. We conclude that RNNs can learn higher-order systematicity in language. However, to increase convergence robustness, additional semantic, behaviorally-grounded sources of information should be incorporated to bootstrap systematicity from semantics.

1 Introduction

This paper examines if and to what extent a conventional Jordan-type neural network (Jordan & Rumelhart, 1992) can develop strong systematicity (Hadley, 1994). Strong systematicity refers to syntactic constraints across the positions in a sentence. For example, any person familiar with English that would accept “Ichiro loves Mary” as being syntactically correct, would also accept “Nancy loves Ichiro” even if he or she never heard of the word “Ichiro” previously. Once “Ichiro” is identified as a noun, the speaker recognizes that it can appear at every noun-suitable position. Thus, strong systematicity refers to a fundamental cognitive bias about the formation of word classes and their grammatical relations.

Hadley pointed out that the connectionist models at that time (Elman, 1990; McClelland, John, & Mcclell, 1990; Pollack, 1990; Smolensky, 1990) acquired an “weakly” systematic representation of word classes. Consider experiments where the connectionist models learn an incomplete subset of sentences generated by the following rules:

$$S \quad ::= \quad N\ V\ N, \tag{1}$$

$$N \ ::= \ \mathtt{n}_1 \,|\, \mathtt{n}_2 \,|\, \cdots \,|\, \mathtt{n}_{\mathtt{N}_\mathtt{N}}, \tag{2}$$

$$V \ ::= \ \mathtt{v}_1 \,|\, \mathtt{v}_2 \,|\, \cdots \,|\, \mathtt{v}_{\mathtt{N}_\mathtt{V}}, \tag{3}$$

where S represents a set of sentences, N and V are groups of words, and $\mathtt{n}_\mathtt{k}(1 \leq k \leq N_N)$ and $\mathtt{v}_\mathtt{k}(1 \leq k \leq N_V)$ are words belonging to the group N and V, respectively. N_N and N_V are the number of words contained in the groups N and V, respectively. According to Hadley, the connectionist models could neither produce nor recognize any sentences whose object is $\mathtt{n}_1$ when $\mathtt{n}_1$ never appeared in the object position in the training data. This result suggests that the network constructed the following unnatural rules:

$$S \ ::= \ N\,V\,N', \tag{4}$$

$$N \ ::= \ \mathtt{n}_1 \,|\, \mathtt{n}_2 \,|\, \cdots \,|\, \mathtt{n}_{\mathtt{N}_\mathtt{N}}, \tag{5}$$

$$N' \ ::= \ \mathtt{n}_2 \,|\, \cdots \,|\, \mathtt{n}_{\mathtt{N}_\mathtt{N}}, \tag{6}$$

$$V \ ::= \ \mathtt{v1} \,|\, \mathtt{v2} \,|\, \cdots \,|\, \mathtt{v}_{\mathtt{N}_\mathtt{V}}, \tag{7}$$

where two separate groups of words N and N' are acquired. Thus, the result cannot be related to our linguistic tendency, although it is not always incorrect in general.

In response to Hadley's paper, two successful connectionist approaches were reported. Phillips (1994) employed a special architecture named tensor-recurrent network tailored for explicitly separating the information about a position and a word. Thus, strong systematicity was realized by enforcing a position-independent representation of a word. Frank (2006) demonstrated that ECHO state networks (Jaeger, 2003) can acquire strong systematicity to a limited extent. In his experiment, the generalization capability of the network was tested after it learned relatively complicated sentences with nested clauses. His results were not perfect, but a fair rate of generalization was obtained considering the complex language. However, it remained obscured how the generalization was represented and if possibly overgeneralizations occurred.

Our approach is closer to Frank's since we also do not enforce a separation between word position and word identity *a priori*. However, we analyze the dynamical structure that is responsible for the re-usage of a word across positions and also control for overgeneralizations. For this purpose, the analysis is conducted from the viewpoint of both the performance and the internal mechanisms in order to clarify the acquired grammatical rules and the underlying dynamics. In particular, we discuss the self-organized mechanisms shared among different positions from the dynamical systems perspective. The identified structures suggest that the self-organization mechanism was able to separate word location information from word identity information—similar to Phillips' pre-wired mechanism. Thus, we conclude that compact word class representations can be found in the stored, word-specific dynamics and may be used as a basis for syntax. Additional semantic information may be incorporated to increase the robustness of the involved learning mechanisms.

2 Learning Model

In our experiment, a single connectionist network learns multiple different sentences in a *deterministic* manner. In the classical Elman study (Elman, 1990), a network learned multiple sentences in a *probabilistic* manner. That is, the output of the network was a probability distribution over the possible next words. This difference is very important to understand our learning method.

Our network is basically a conventional three-layer Jordan-type neural network (Fig.1). The input layer consists of 11 input nodes and 30 context input nodes. Each input node corresponds to a specific word (6 nouns, 4 verbs, and a dummy word that designates the beginning of a sentence). A word is encoded as a localist vector: only the node that corresponds to the word is activated (0.5) and the others are inactivated (−0.5). The hidden layer contains 30 hidden nodes. The output layer has 10 output nodes and 30 context output nodes. Each output node corresponds to a specific word in the same manner as the input nodes. Context output nodes are connected to the context input nodes through the context loop. The output function tanh() is applied in the nodes of the hidden and output layers, effectively constraining output values to the range [−1.0, 1.0].

The network is supervised to generate a sentence word by word. It predicts the next word in its output nodes from a word presented in its input nodes. The sentence to be generated is determined by an initial activation vector of the context input nodes (Nishimoto & Tani, 2004). These vectors

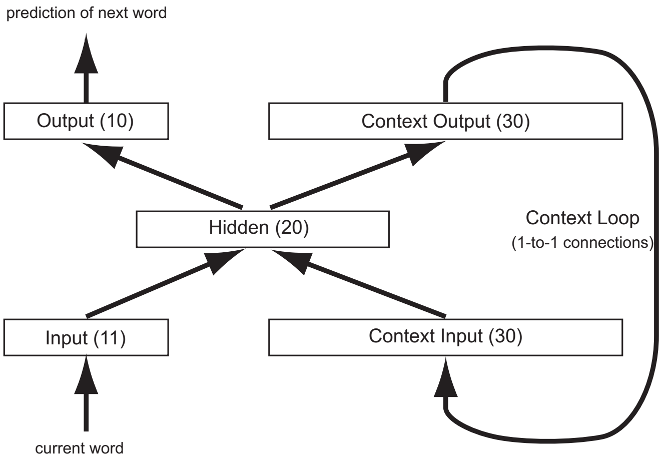

Figure 1: **The architecture of the learning model.** A box represents a neural layer with the indicated number of neurons.

self-organize through the learning process, as explained below. After setting the context input values, the network generates the first word of the sentence given the context and the dummy word input values. The context values are then overwritten by the context output value via the context loop. Next, the produced word (closed-loop) or the sampled word (teacher forcing) determines the next input and the process continues producing the second word, etc. The network generates words until it reaches the last time step, which is detected when all context output nodes become zero[1].

The network learns all the sentences included in the supervising data in batch learning mode. The learning algorithm is basically the conventional back-propagation through time (BPTT) algorithm. Both, the connection weights $\boldsymbol{\theta}$ and the context representation vectors (CRVs) $\boldsymbol{u}$ are subject to optimization. These two sets of parameters are optimized by using the same algorithm with respect to different error functions. The connectivity $\boldsymbol{\theta}$ is optimized with respect to the summation of the output error E over all the words of all the sentences that are included in the training data $\mathcal{S}$. Meanwhile, the vector $u_s \in \boldsymbol{u}$, which represents a sentence $s \in \mathcal{S}$, is optimized with respect to the particular output error E_s of this sentence. Both error functions are defined in terms of the output error e_s of each generated word, which is the Euclidean distance between the desired and actual output as follows:

$$
\begin{aligned}
E(\boldsymbol{\theta}, \boldsymbol{u}) &= \sum_{s \in \mathcal{S}} E_s(\boldsymbol{\theta}, u_s), && (8)\\
E_s(\boldsymbol{\theta}, u_s) &= \sum_{t=0}^{l_s-1} e_s(t; \boldsymbol{\theta}, u_s), && (9)\\
e_s(t; \boldsymbol{\theta}, u_s) &= \|\hat{y}_s(t) - y(t; \boldsymbol{\theta}, u_s)\|^2, && (10)
\end{aligned}
$$

where l_s is the length of a sentence s, $\hat{y}_s(t)$ is a localist vector representing the t-th word of the sentence, $y(t; \boldsymbol{\theta}, u_s)$ is the actual output of the network at time step t under the connectivity $\boldsymbol{\theta}$ and the CRV u_s. The output is calculated according to the architecture of the network depicted in Fig.1.

At the beginning of learning, the connectivities are initialized with small random values and all CRVs are reset to $\mathbf{0}$. As learning proceeds, the common information among the sentences is learned within the connection weights while the information specific to each single sentence is learned in the respective CRV, which self-organizes in the context nodes at the initial time step of each sampled sentence.

[1] The context output values at the last time step are supervised to be zero in the learning phase.

3 Experimental Procedure

In the experiments, the above-mentioned connectionist network learns six different sets of sentences. The sentences are generated according to the syntactic rules expressed above in (1) – (3). All the training sets are designed to clarify to what extent the network can conceive strong systematicity. The first and second set consists of sentences that are produced by rules just as the above-mentioned rules, where $N_N = 6$ and $N_V = 4$. In the first set, a noun $\mathtt{n}_1$ never appears at the agent position. Similarly, in the second set no sentence has noun $\mathtt{n}_1$ as an object. For convenience, we denote the former set by "$N'VN$" and the latter set "NVN'". Based on this convention, the remaining four sets of sentences are labeled as $VN'N$, VNN', $N'NV$, and $NN'V$. Every training set contains 65 out of the possible 120 sentences. During testing, the generalization capability was tested on the 144 correct sentences, including the inhibited word in the N' set, as well as on other ungrammatical word sequences, such as NNN.

Each experimental session consists of two subsequent phases: learning and testing. In the learning phase, the network learns from all the provided 65 sentences in batch mode adjusting connectivity weights and the CRVs 20,000 times in total based on BPTT. The learning rates are 0.0012 for connectivity wight adjustments and 0.0004 for the CRVs. After successful learning, it is tested to what extent the networks can re-generate all the 144 possible sentences including the previously unseen ones. To evaluate the reproduction capabilities, the learned sentences are re-generated based on the respective CRVs obtained in the learning phase. A sentence is judged as being generated correctly if the network generates all the words in the sentences in an appropriate order. At each time step, an output node that corresponds to a right word must be activated at the highest rate among all the output nodes.

To evaluate generalization to unlearned sentences $s \notin \mathcal{S}$, their CRVs u_s are computed by using the learning algorithm without updating the connectivity $\boldsymbol{\theta}$. The generalization performance is evaluated considering (1) if the CRVs of unlearned but syntactically correct sentences are well-embedded and (2) to what extent the network distinguishes syntactically correct sentences from incorrect ones. This analysis will reveal the rules that the network constructed from the provided positive examples. In the ideal case, the generation of any illegal word sequence would undergo higher error than all the syntactically correct sentences. We measure the defectiveness of a sentence E'_s based on the output error of the most defective word in the sentence s as follows:

$$E'_s(\boldsymbol{\theta}, u_s) \quad = \quad \max_{t \in \{0,1,2\}} \sqrt{e_s(t; \boldsymbol{\theta}, u_s)}, \tag{11}$$

where e_s is the output error of each single word defined in (10). The precision of the separation is evaluated based on the distribution of E'_s over all the possible 3-word sequence.

4 Results and Analysis

The network could reconstruct the appropriate syntax rules in three out of the six training sets. In the $N'VN$, $VN'N$, and $N'NV$ conditions, the network worked well with regard to the above-mentioned two aspects, that is, all the correct sentences were generated appropriately and were clearly separated from the illegal ones in terms of the output error. The result of the $N'VN$ condition is presented in detail below. However, the network failed to reconstruct the underlying rules in NVN', VNN', and $NN'V$ conditions. No sentences in which $\mathtt{n}_1$ appears at the N'-position could be generated. The successful conditions are characterized by an N' preceding an N, apparently due to a learning bias from the second, full set of nouns to the first, restricted set of nouns.

4.1 $N'VN$ Condition

Five sessions of the experiments were conducted in the $N'VN$ condition. In each session, the network was initialized with different random values. A fixed training set was employed. As mentioned above, sentences starting with $\mathtt{n}_1$ were eliminated from the training set.

The network could re-generate all the possible sentences in four out of the five sessions. In the remaining one session, the network failed to re-generate four sentences that included the left-out noun but could generate all $\mathtt{n}_1VN$ sentences. In all successful cases, illegal sentences were rejected well. False positives were less than 10 percent of possible illegal sentences, even in the worst case

Table 1: **Separation of correct sentences from incorrect sequences of words:** For each of the four successful sessions, the table shows the worst error for correct sentences (E'_{max}) as well as the number of sequences that yielded error values below E'_{max} for each type of illegal word sequence.

Session	E'_{max}	# of false positives							
		NNN	*NNV*	*NVV*	*VNN*	*VNV*	*VVN*	*VVV*	total
1	0.33	3	0	0	0	0	0	0	3
2	0.59	5	6	0	22	16	35	0	84
3	0.53	4	4	0	24	11	19	0	62
4	0.38	0	0	0	0	0	0	0	0

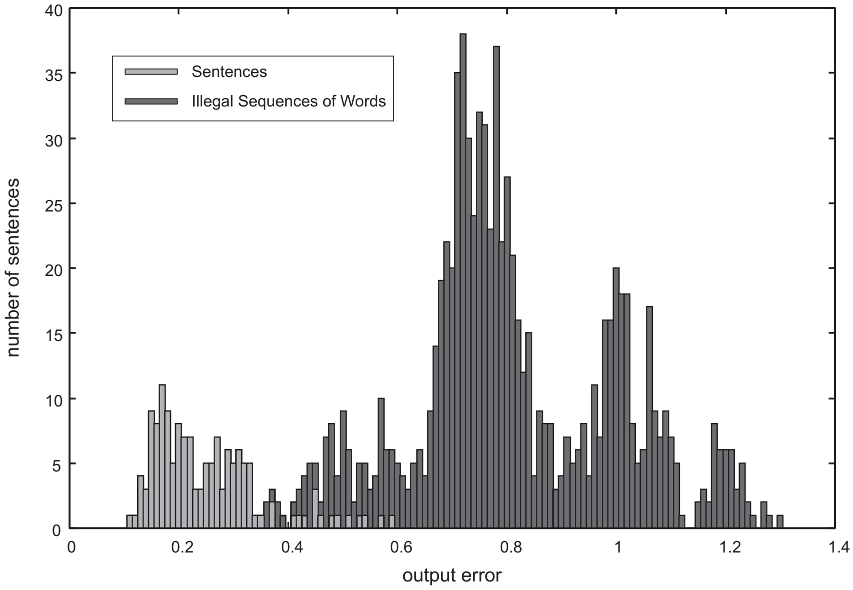

Figure 2: **Sentence- and word sequence-specific distribution of output errors**

as shown in Table 1. The major fraction of the false positives shared a fragment that consisted of a verb followed by a noun. It seems that the fragment was generated easily since it was contained in the correct sentences as well.

The error distribution of the worst case (the second session) is depicted in Fig.2. In this case, 15 unlearned sentences had higher errors than an illegal sequence of words, 14 out of the 15 began with $\mathtt{n}_1$. Thus, the clear separation critically depends on the accurate re-use of $\mathtt{n}_1$ across the positions.

5 Discussion

As mentioned in the introduction, strong systematicity is explained from two different points of view: (1) the re-use of words across the positions and (2) the acquisition of syntax and word classes. In approaches based on symbolic representations, the former is explained or realized in terms of the latter. Contrary to this, in sub-symbolic approaches, including ours, the latter is concluded based on the observation of the former. We now discuss to what extent our network has acquired a syntax generalized over classes of words and what appear to be the current restrictions of the taken approach.

5.1 Sub-symbolic Mechanisms Underlying the Word Class

Several results enable us to narrow down the list of possible sets of rules that explain the observed generalization performance. Due to the rejection of the false positives, we can exclude the possibility that the network learned the following set of rules:

$$S ::= W\ W\ W, \tag{12}$$

$$W ::= \mathtt{n}_1 | \cdots | \mathtt{n}_6 | \mathtt{v}_1 | \cdots | \mathtt{v}_4, \tag{13}$$

which are completely different from the original rules and simply state that any word may appear at any position in a three-word sentence.

Since the network behaved as if it were following the syntax rules shown in (1) – (3), we now investigate the sub-symbolic mechanisms underlying the syntax further for the $N'VN$ condition. To do so, we analyze the activity development of the activation vectors in the context units while a sentence is observed. Figs.3(a)–(e) show the activation vectors of the context input units for each of the 144 sentences. Fig.3(a) depicts the distribution of the initial vectors with respect to the agent of the corresponding sentences. Similarly, Fig.3(b) depicts the distribution of the vectors just before generating an object with respect to the object. In both figures, 30 dimensional activation vectors are projected into an identical two-dimensional plane. The plane is determined by using conventional principal component analysis (PCA). In Figs.3(c) and (d), the activation vectors are averaged with respect to the agent and object, respectively. It should be noted that almost congruent geometric arrangements can be observed in both figures. The congruency across the positions of a sentence indicates that the structure of the word class N became approximately position independent.

The clusters in Fig.3(a) are scattered more than in Fig.3(b) since the initial context vector has to encode more information about upcoming words. At the initial time step, a context vector CRV represents an entire sentence to be generated. As the network produces a sentence word by word, it forgets the produced words and focuses on the remaining words. The forgetting and focusing are realized in terms of a contraction and expansion map, respectively. Fig.3(e) shows the representation of the object in the initial context activations. In the figure, the initial context vectors shown in Fig.3(a) are averaged with respect to the object instead of the agent. Further analyses show that almost the same geometric arrangement can be seen among every six sentences sharing an identical agent and verb (not shown). This local sub-structure is unfolded into the global structure shown in 3(d) through the above-mentioned expansion map.

5.2 Currently Known Restrictions

Besides the restriction that generalization was observed only from the later (object) to the earlier (agent) noun occurrence, we are aware of two other current system limitations. First, the network fails to learn training data where more than one noun are eliminated from a specific position. Thus, the generalization capabilities are currently limited. However, it has to be kept in mind that the encoding of each noun is currently completely independent so that the scalability without further information is inevitably limited.

Second, the correct formation of the word class may be precluded by a strongly biased training data. For example, the network considered n_1 as a special noun in the training set, which contains all the possible sentences other than sentences beginning with n_1. This implies that the words are classified based on similarities between occurring contexts. If the sampled sentences differ significantly in the occurrence of nouns as agent and object, it is difficult for the network to interpret that both positions share the identical distribution of words.

6 Conclusions and Future Directions

The experimental results confirm that a conventional Jordan-type neural network can learn the strong systematicity based on purely lexical information to a limited extent. However, the very simplistic encoding, which does not provide any information about semantic similarities apart from the syntactic occurrence, is probably not sufficient to enforce unstructured neural networks to develop more general forms of strong systematicity. Thus, it will either be necessary to constrain the neural architecture to enforce the separation of location and content information, or to incorporate semantic information, as pointed out by (Phillips, 2000).

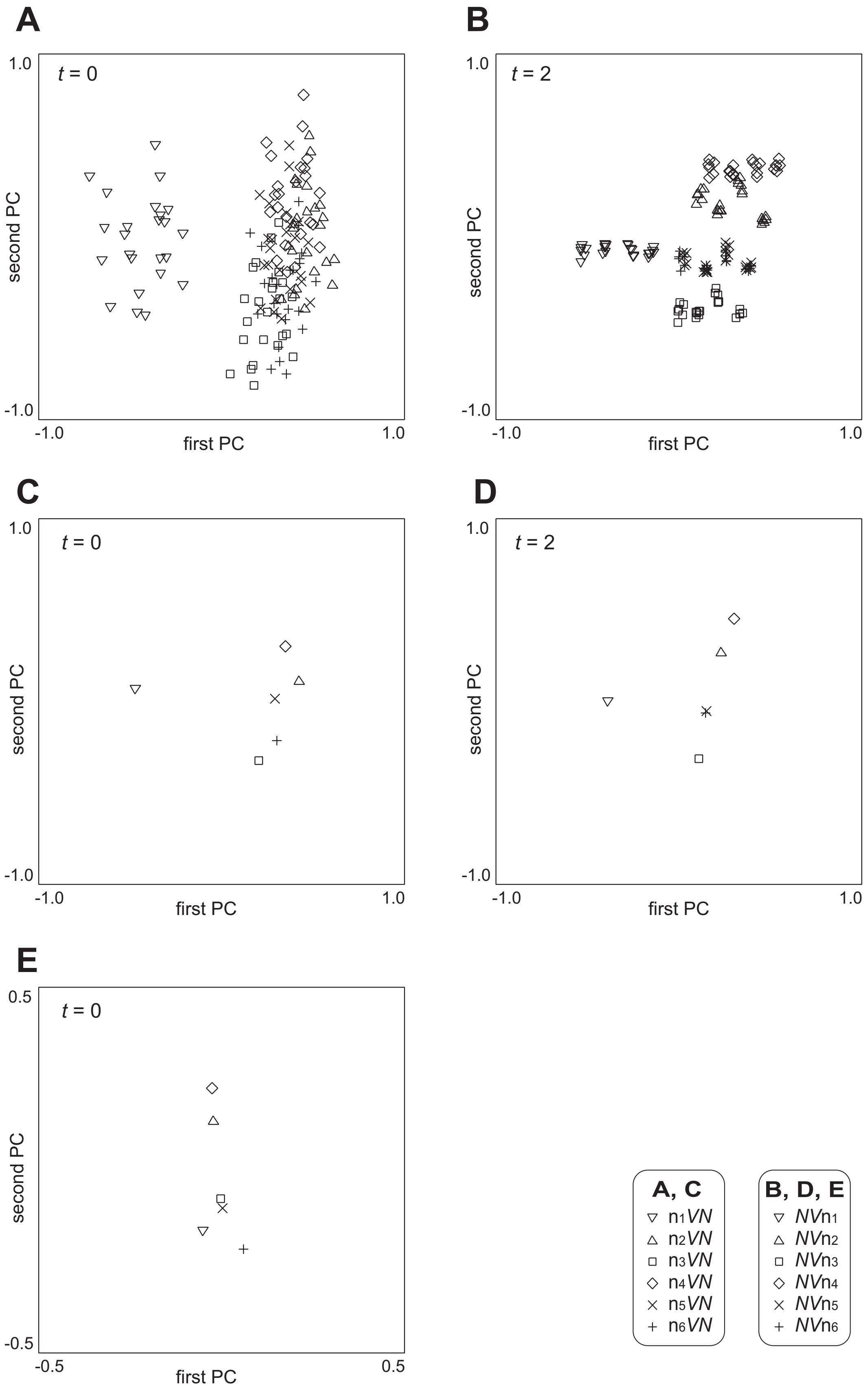

Figure 3: **Geometric distributions of the context activation vectors.**

Our most pressing future research direction lies in the improvement of the generalization performance and the inclusion of semantic information to realize this improvement. Essentially, we are planning to investigate how semantic, sensorimotor interaction structures may facilitate the acquisition of word classes. Similar to the computational model of semantic bootstrapping in the acquisition of syntax (Dominey, 2006), this study will be associated with a sub-symbolic model that conceptualizes actions of the robot in a compositional manner (Sugita & Tani, 2008). Thus, the lexical categories of words may be grounded in the semantic structure, potentially giving further clues on the difference between nouns and verbs as well as on the similarity structures within classes of nouns and verbs.

References

Dominey, P. (2006). From holophrases to abstract grammatical constructions: insights from simulation studies. In E. Clark & B. Kelly (Eds.), *Construction in Acquisition* (pp. 137–162). Stanford CA: CSLI Publications.

Elman, J. (1990). Finding Structure in Time. *Cognitive Science*, *14*, 179–211.

Frank, S. L. (2006). Strong Systematicity in Sentence Processing by an Echo State Network. In S. Kollias, A. Stafylopatis, W. Duch, & E. Oja (Eds.), *Proceedings of the 16th International Conference on Artificial Neural Networks (ICANN 2006)* (pp. 505–514). Berlin: Springer-Verlag.

Hadley, R. (1994). Systematicity revisited: reply to Christiansen and Chater and Niklasson and van Gelder. *Mind and Language*, *9*, 431–444.

Jaeger, H. (2003). Adaptive nonlinear system identification with echo state networks. In S. T. S. Becker & K. Obermayer (Eds.), *Advances in neural information processing systems 15* (pp. 593–600). Cambridge, MA: MIT Press.

Jordan, M., & Rumelhart, D. (1992). Forward models: supervised learning with a distal teacher. *Cognitive Science*, *16*, 307–354.

McClelland, L., John, M. F. S., & Mcclell, J. L. (1990). Learning and applying contextual constraints in sentence comprehension. *Artificial Intelligence*, *46*, 217–257.

Nishimoto, R., & Tani, J. (2004). Learning to Generate Combinatorial Action Sequences utilizing the Initial Sensitivity of Deterministic Dynamical Systems. *Neural Networks*, *17*(7), 925–933.

Phillips, S. (1994). Strong systematicity within connectionism: The tensor-recurrent network. In A. Ram & K. Eiselt (Eds.), *Proceedings of the sixteenth annual conference of the cognitive science society* (pp. 723–727). Lawrence Erlbaum.

Phillips, S. (2000). Constituent similarity and systematicity: The limits of first-order connectionism. *Connection Science*, *12*(1), 45–63.

Pollack, J. (1990). Recursive Distributed Representations. *Artificial Intelligence*, *46*, 77–105.

Smolensky, P. (1990). Tensor Product Variable Binding and the Representation of Symbolic Structures in Connectionist Systems. *Artificial Intelligence*, *46*, 159–216.

Sugita, Y., & Tani, J. (2008). A sub-symbolic process underlying the usage-based acquisition of a compositional representation: Results of robotic learning experiments of goal-directed actions. In *Proceedings of the 7th IEEE International Conference on Development and Learning (ICDL 2008)* (pp. 127–132).