TECHNICAL REPORT NO. 300

Networks that Learn Phonology

by

Michael Gasser and Chan-Do Lee

December 1989

COMPUTER SCIENCE DEPARTMENT
INDIANA UNIVERSITY
Bloomington, Indiana 47405-4101

Networks that Learn Phonology

Michael Gasser
Chan-Do Lee

Computer Science Department
Indiana University
Bloomington, IN 47405, USA

Abstract

Natural language phonology presents a challenge to connectionists because it is an example of apparently symbolic, rule-governed behavior. This paper describes two experiments investigating the power of simple recurrent networks (SRNs) to acquire aspects of phonological regularity. The first experiment demonstrates the ability of an SRN to learn harmony constraints, restrictions on the cooccurrence of particular types of segments within a word. The second experiment shows that an SRN is capable of learning the kinds of phonological alternations that appear at morpheme boundaries, in this case alternations like those occurring in the regular plural forms of English nouns. This behavior is usually characterized in terms of a derivation from a more to a less abstract level, and in previous connectionist treatments (Rumelhart & McClelland, 1986; Plunkett & Marchman, 1989) it has been dealt with as a process of yielding the combined form (plural) from the simpler form (stem). Here the behavior takes the form of the more psychologically plausible process of the production of a sequence of segments given a meaning or of a meaning given a sequence of segments. This is accomplished by having both segmental and semantic inputs and outputs in the network. The network is trained to auto-associate the current segment and the meaning and to predict the next phoneme.

1. Connectionist Phonology

Recently, George Lakoff (1989a, b) and David Touretzky (1989) have called the attention of the connectionist community to phonology as a fruitful problem domain. Phonology is generally viewed as symbolic, rule-governed behavior; thus it presents the kind of challenges for connectionist models that have been brought forward by Fodor and Pylyshyn (1988) and Pinker and Prince (1988). In addition, phonology is constrained in ways which do not always have natural explanations in conventional accounts. If connectionist networks can model behavior which is apparently governed by phonological rules, make this knowledge generally available to the linguistic system (rather than having it embedded in one isolated component), and have the constraints fall out naturally, then this will constitute strong evidence for the computational power of connectionism.

Connectionist phonology is as yet in its infancy. Rumelhart and McClelland (1986) showed that a simple pattern associator could learn the regular phonological alterations in the English past tense morpheme, as well as a number of irregular forms. This model has been criticized on many grounds (Pinker & Prince, 1988), some of them answered by Plunkett and Marchman (1989) in a later version of the model. For our purposes, however, both the original model and Plunkett and Marchman's improved model suffer from the fact that neither implements a process which actual language users need to carry out.[1] These networks turn a verb stem into a past tense form, whereas language users have the task of turning a verb form into a semantic characterization of the verb or a semantic characterization into a verb form. This is not a model of language processing.

Lakoff's (1989a, b) **cognitive phonology** is a connectionist alternative to generative phonology. Touretzky and Wheeler (Touretzky, 1989; Touretzky & Wheeler, 1989) have developed an implementation of Lakoff's ideas, making modifications along the way. Lakoff's theory, and its extension by Touretzky and Wheeler, replaces the long serial derivations of standard generative phonology with a psychologically more plausible approach making use of parallelism and constraints on possible constructions. Yet cognitive phonology shares several key features with standard models. First, it posits abstract levels and underlying forms. While not arguing against the psychological reality of such constructs, we are interested in where they may have arisen in the system in the first place, assuming they were not there to start out with. We feel that connectionism is ideally suited to investigating the acquisition of phonology. Second, cognitive phonology is a competence model; it does not treat the role of phonology in the concrete processes of perception and production. Our concern is with phonological performance.

Our goal then is to to train a network that does not have the benefit of underlying forms and abstract levels to perform tasks similar to those that real language users are faced with. We have started with a relatively constrained network architecture, one in which feedforward connections are supplemented by limited feedback connections. The paper describes two simulations making use of this architecture. In both cases, the basic task that the network is trained on is a simple one, that of predicting the next phonological segment in a word.

2. Simple Recurrent Networks

Most temporal processes require some sort of short-term memory (STM). That is, a system cannot know how to behave on the basis of only the current input; the context of that input is also relevant. Early connectionist approaches to temporal processes made use of "windows": the input to the system consisted of whole sequences of primitive events of a fixed length, for example, 7 phonemes in a system concerned with phonological input such as NETTALK (Sejnowski & Rosenberg, 1987). The STM in these systems is just the fixed set of previous inputs. A related approach uses time-delay connections: only one event is presented at a time, but each input unit has a set of connections emanating from it with different time delays, making it possible for a higher layer of units to have access to inputs that appeared on earlier time steps. Because the time delays are fixed and because the STM consists of unanalyzed input events, this approach is quite similar to the window approach.

Recently there has been considerable interest in models with more interesting sorts of STMs, models in which recurrent connections permit aspects of the system at one time step to influence the behavior of the system at the next time step. There are several variants; in this paper we consider only the **simple recurrent network** (SRN) developed by Elman (1988, 1989) as a modification of a related architecture due to Jordan (1986). The basic structure of an SRN is shown in Figure 1. An SRN is based on a simple feedforward pattern associator consisting of an input layer, one or more hidden layers, and an output layer of units. One set of input units represents a single input event. What gives the network its STM is a further set of input units, usually referred to as the "context" layer, which copy the activations on one of the hidden layers from the previous time step. Thus on any given time step, the network has access not only to the current input event but also to an analyzed version of its previous input. Because this previous input also included the STM pattern, there is the potential for the network to make use of information at various time delays. The STM is limited only by the amount of information that can be stored in the hidden layer.

---

[1] In both cases the authors were aware of this limitation.

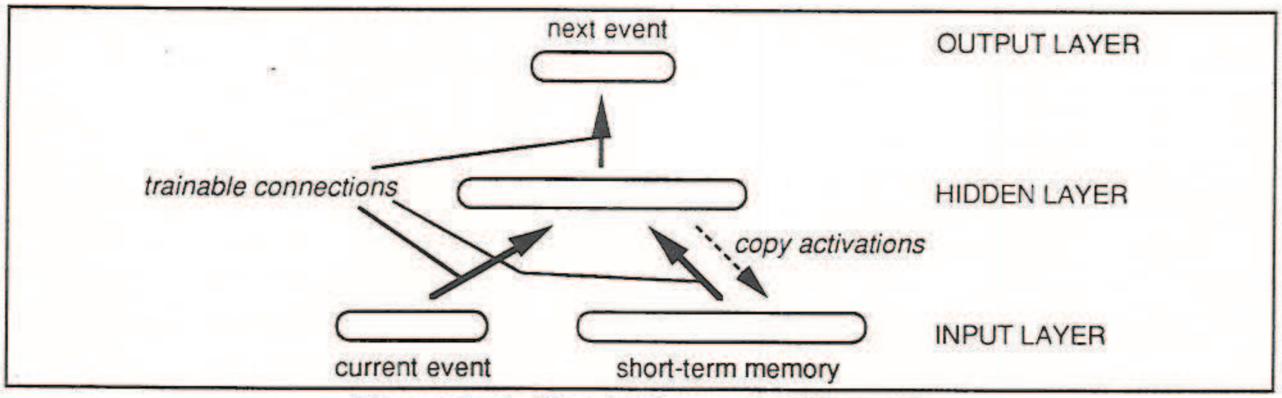

Figure 1: A Simple Recurrent Network

Elman and others who have used this architecture (e.g., Servan-Schreiber, Cleeremans, & McClelland, 1989) train it on prediction tasks. That is, given an input sequence, the network is presented on a given time step with one event in the sequence and expected to generate the next event on its output layer. Backpropagation (Rumelhart, Hinton, & Williams, 1986) is used to train the network. If the input sequences exhibit regularities across successive events, or events separated by intervening events, the network has the potential to detect these regularities and use them to make predictions about novel sequences.

In the simulations described in this paper, there is one modification to the usual approach. In addition to being trained to do prediction, the networks are also trained to generate the current input event. That is, there are two sets of output units, one for the current and one for the next event. This auto-association forces the network to distinguish the different input event patterns on the hidden layer, a prerequisite to the system's making use of the hidden layer as an STM (Servan-Schreiber, Cleeremans, & McClelland, 1989).

3. Phonological Knowledge

3.1. Well-Formedness Constraints

For our purposes, it is useful to classify the phonological knowledge that speakers and hearers have into two categories. On the one hand, there are the well-formedness constraints that define the basic shape of syllables and words. An example is a so-called **harmony** constraint. Such constraints force two or more segments or sequences of segments in the same word to agree along particular dimensions. For example, Turkish is a language that is characterized by vowel harmony. With some exceptions, if the first syllable of a Turkish word is a front, back, or unrounded vowel, then succeeding vowels are also front, back, or unrounded respectively, while an initial rounded vowel may be followed by either rounded and close or unrounded and open vowels.

In order to train a network to learn regularities of this type, we could simply expose it to large numbers of words. The question is whether a simple architecure like the SRN is capable of learning regularities such as vowel harmony which operate across sequences of segments of different lengths. For example, two vowels related by a harmony constraint might be separated by varying numbers of consonants in a given language. The network needs to be able to predict a vowel of the appropriate type in all such cases. Experiment 1, described below, is designed to test this ability of SRNs.

3.2. Phonological Processes

On the other hand, there are aspect of phonological behavior that seem to require rules to characterize them. An "underlying" segment or sequence of segments[2] is said to undergo a change in a particular environment, yielding a form which is closer to the "surface". Invariably there is an interaction between these processes and morphology because it is only on the boundaries between morphemes that it makes sense to speak of segments changing. Note that it is not necessary to view these aspects of language as processes or rules. The process view implies that there is a "derivation" from a more abstract to a less abstract level of processing. Another possibility is that the forms are in some way generated directly, that the transformations associated with the rules are built into the associations between forms and meanings.

Consider a familiar example, variation in the English regular plural morpheme. There are three variants, /s/, /z/, and /ɪz/. If we assume that /z/ is the underlying form of the morpheme, the distribution of the variants can be characterized by the following ordered phonological rules:

(1) z -> ɪz / [+sibilant] __
(2) z -> s / [-voice] __

Note that these are phonological and not purely morphological rules because they apply to a variety of distinct morphemes, including the third person singular present verb suffix (Pinker & Prince, 1988).

---

[2] Or (in non-linear analyses) an element at a tier other than the segmental level.

The learning of a phonological process such as this requires that the system combine two input morphemes. Phonologists typically view this as a matter of starting with the underlying forms of the two morphemes and yielding the combined, surface form. This view is reflected in the rules above. However, since we presuppose only surface phonological forms, this is not an option for our system. Instead we must train the network on the associations between a surface form and a separate lexical, or semantic, representation of the morphemes themselves. This allows us later to test the system on knowledge of the rule by, for example, giving it the meaning of a combined form and asking it to generate the corresponding form.

For example, for the English plural, we would like to train the network on pairs like the following:

(3) LEAK + SINGULAR -> /lik/
(4) LEAK + PLURAL -> /liks/
(5) PACK + SINGULAR -> /pæk/

and then test it on a pair like the following:

(6) PACK + PLURAL -> ??,

where the items in small capitals represent meanings.

However, this solves the problem in only the production direction. Our system should also arrive at meanings given forms. That is, we would like a system which has been trained on (7), (8), and (9) to be able to do (10):

(7) /lik/ -> LEAK + SINGULAR
(8) /liks/ -> LEAK + PLURAL
(9) /pæk/ -> PACK + SINGULAR
(10) /pæks/ -> ??

To learn both how to produce and perceive phonological sequences, the system needs to be trained on bidirectional associations between meanings and forms.

4. Experiment 1: Extracting Well-Formedness Constraints

4.1. Inputs and Training Regimen

Experiment 1 investigates the ability of an SRN to extract regularities from purely segmental input. Specifically, we were interested in whether a network could learn a simple harmony rule, that is, whether following training it would be able to predict those features of the second vowel in a presented word which are constrained by the first vowel in that word.

Our inputs consisted of sequences of phonological segments representing words in an artificial language. Segments were presented over a set of 8 input units, each representing a binary phonetic feature. The phonological inventory of the language consisted of 9 consonants and 5 vowels. Each word was of the form (C)V(C)CV; medial consonant clusters consisted either of a nasal plus a stop or fricative with the same place of articulation as the nasal consonant or a fricative plus a stop. The vowel harmony rule constrained vowels within a word to take the same values for the features BACKNESS and FRONTNESS; /i/ and /e/ were +FRONT, -BACK; /u/ and /o/ –FRONT, +BACK; and /a/ –FRONT, –BACK. Thus the following are legal words in the language: *pafa*, *mixte*, *osu*, *kempe*, and *nunu*.

The network used in this simulation is illustrated in Figure 2. There were 24 units in the hidden layer, which was copied following back-propagation on each time step to the set of STM input units. There were 8 output units for the current segment and 8 for the next segment.

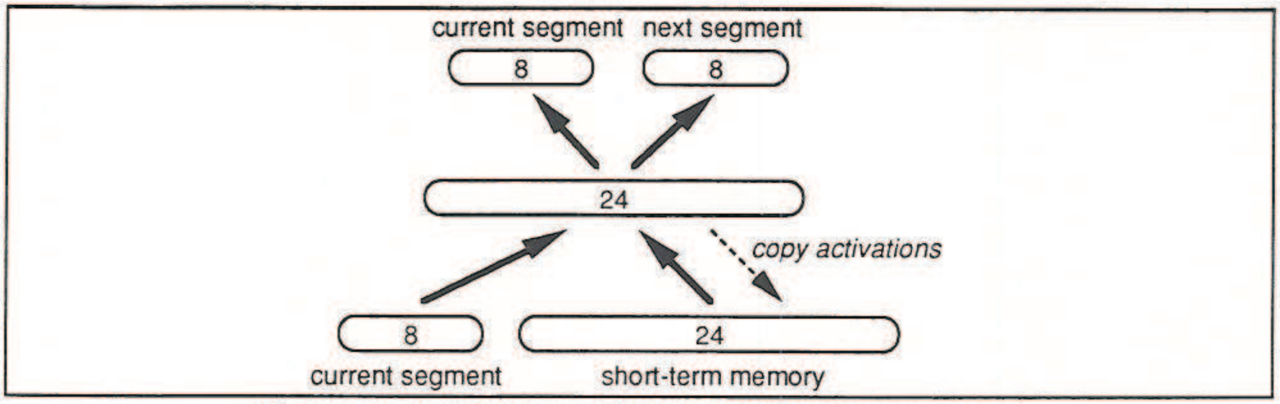

Figure 2: Architecture of Network for Experiment 1

A set of 500 training words and 25 test words was generated randomly (out a possible total of 2160 words). Training words were selected randomly from the training set for presentation to the network. Each word began with a word boundary, consisting of a "segment" with an activation of .25 on all the segment input units. The network was trained on 2000 words in all.

4.2. Results

To evaluate the network, we need only be concerned with the output units representing the FRONT and BACK features on the predicted segment units for the second vowel in each word. The activations of these units should tend to match those for the same features on the first vowel in the word. However, since the second vowel may follow one or two consonants, the network can in general not be certain that a vowel follows a single medial consonant (unless this is a stop). In the cases where either a consonant or a vowel could follow, we hoped that the network would show signs of predicting both a consonant and a vowel with the appropriate features. That is, the activations on the FRONT and BACK output units should still reflect the harmony rule but to a lesser extent than in the case when the vowel necessarily follows.

Therefore, the following criteria were used for evaluating the network's behavior. For words containing a consonant which had to be followed by a vowel (the second consonant in a medial pair or the single consonant if this is a stop), the prediction output units for the FRONT and BACK features at the presentation of this consonant were examined. Any value over .5 was treated as 1, any value under .5 as 0. For other words the relevant output units at the presentation of the single medial consonant were examined. In this case any value over .25 was treated as 1, any value under .25 as 0.

Given these criteria, the network succeeded on all of the 25 test words, indicating that the harmony rule had been learned for all of the variant word shapes.

5. Experiment 2: Learning a Phonological "Rule"

5.1. Architecture

Experiment 2 tests the ability of an SRN to learn behavior which can be described in terms of a change from an underlying to a more surface form, that is, a process normally thought of as a rule. Specifically, we were interested in the acquisition of the rule behind the variation in the English plural morpheme. As we have seen, this requires the learning of associations between meaning and form.

Figure 3 shows the architecture of a network designed to perform this task. This is similar to the network used in Experiment 1, except for the addition of meaning units on the input and output layers. The meaning inputs and target meaning outputs are constant throughout the presentation of a word. Thus the network is trained on auto-association for both meaning and segments and also on prediction for segments.

This network has the capacity to associate form with meaning as well as form with form and meaning with meaning. To test the network in the perception direction, that is, from form to meaning, the appropriate units are turned on for the sequence of segments in the word, and the meaning units (STEM and/or NUMBER) are set initially to default values. As the word is presented, the output meaning units should begin to take on the correct activations. To test the network in the production direction, that is, from meaning to form, the meaning input units are set to the appropriate values, and the NEXT SEGMENT output units are examined on the appearance of each input segment. Either the appropriate input segments may be presented, or the segment input units can be set to the values predicted on the NEXT SEGMENT units on the previous time step.

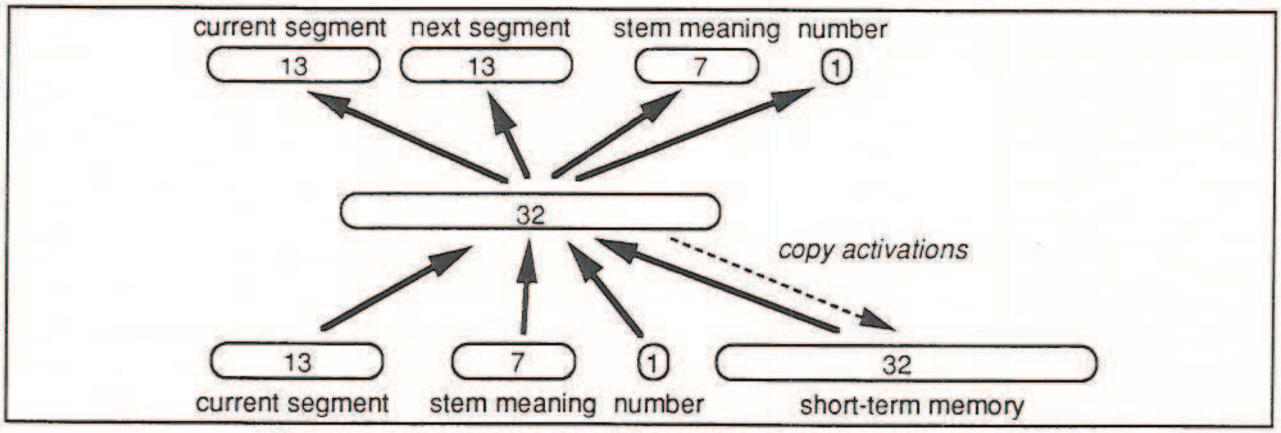

Figure 3: Architecture of Network for Experiment 2

5.2. Inputs and Training Regimen

The training set for this set of simulations consisted of a set of 20 one-syllable English nouns. Twelve of these were designated "training" words, the other 8 "test" words. The network was trained on both the singular and plural forms of the training words and only the singular forms of the test words. Words were presented one segment at a time. Segments appeared on 13 input units representing

Chomsky-Halle phonetic features (Chomsky & Halle, 1968). Each word ended in a word boundary pattern consisting of all zeros.

Stem meanings were represented by arbitrary patterns consisting of 3 units on within a group of seven stem meaning units. A single unit represented number; 1 signified plural, 0 singular. The input and target meaning patterns remained constant throughout the presentation of a word (except in the simulation involving uncertainty described below).

We initially trained the network on the basic auto-association and prediction task. That is, the network was always given complete words and meanings as inputs. This proved to result in the desired learning in the production direction, but not in the perception direction. Therefore, we retrained the network according to the following regimen: On 4 out of every 5 words, the network saw complete words and meanings as before. On 1 out of every 5 words, the input number was treated as unknown. That is, the number unit was set word-initially to the default value of .5 and later to the value that it took on on the previous time step. For example, the inputs for the word *hat* for the "unknown number" case consisted of the following:

segment	stem	number	
h	HAT	.5	(target = 1)
æ	HAT	previous NUMBER output	(target = 1)
t	HAT	previous NUMBER output	(target = 1)
s	HAT	previous NUMBER output	(target = 1)

5.3. Results

To test the network's performance in the production direction, we set the stem meaning input units to the meaning of a particular stem and the number unit to plural (1), gave the network the appropriate segments for the stem in sequence and them examined the prediction output units at the point where the plural morpheme should appear. We converted each output pattern to the nearest phoneme (in Euclidian distance). The network predicted the correct segments for all of the training words and test words.

To test performance in the perception direction, we proceeded as in the "unknown number" training trials. That is, we gave the network the sequence of input segments for the singular or plural form of a word, set the stem units to the appropriate pattern, and set the input number unit initially to .5. At the presentation of each new segment, the number unit was then set to whatever activation the corresponding output unit had. We then examined the behavior of the output number unit following the appearance of either the appropriate plural morpheme or a word boundary on the input segment units. For the training words, the output number unit fluctuated around .5 until the relevant information appeared on the input segment units. Then it correctly turned on or off according to whether the plural morpheme or a word boundary appeared. For the test words, the output number unit remained near 0 before the appearance of the relevant input information. This is not surprising since these words were trained only in the singular form. When the plural morpheme or word boundary appeared, the output number unit behaved appropriately for 7 of 8 test items. In the one error, the network treated a novel plural noun as singular.

We have also been able to train networks to produce entire words given only their meanings and to generate complete stem meanings given only segmental input, but this has not generally been possible for novel combinations, that is, for the plural forms of test items.

Thus while the network does not yet achieve all we would like, the results indicate that it has the ability to learn the relevant rule-like behavior in a manner that makes it usable in both perception and production.[3]

6. Discussion and Future Work

Experiment 1 demonstrates the possibility of extracting within-word regularity from phonological input. In addition to harmony rules of the type acquired by the network in this simulation, there are many other constraints on the environments in which phonemes may appear. The network in the simulation, for example, also learned that in our artificial language, a fricative following a vowel may be followed by a stop or a vowel. Linguists have focused on absolute regularities of this type, but languages certainly exhibit statistical regularities as well, and networks would seem to provide an ideal means of detecting them.

Experiment 2 demonstrates the potential for networks to acquire phonological processes without pre-wired levels or categories. It is important to note how this simulation differs from the work of Rumelhart and McClelland (1986) and Plunkett and Marchman (1989) on the acquisition of English past tense. Note that the past tense rule itself is virtually identical to the plural rule studied here. First, the present model deals with aspects of actual processing, that is, getting from meaning to form and from form to meaning, rather than with an abstract form-to-form mapping that need not correspond to any part of processing. Second, the network does not constitute an isolated module devoted to the representation of a single category of morpheme. Rather, many of the same connections could presumably be used in the representation of other morphemes and phonological processes, though this would require additional input units on the semantic end (see the discussion in the next section). Third, the use of a recurrent architecture avoids the window approach to phonology characteristic of the earlier models, one of the features of these models which justifiably attracted a good deal of criticism (Pinker & Prince, 1988).

However, there is still much to be done before it has been shown that this approach is capable of acquiring even the simple plural rule in an adequate manner. First, it must be possible for the network to represent irregular forms alongside the regular ones, as Rumelhart and McClelland did in their original past-tense model. Second, it must be shown that the knowledge about the plural morpheme transfers easily to knowledge about the third-person singular present form of verbs, which has exactly the same set of phonological variants. That is, it should be significantly easier for a network of this type to learn the forms of the third person singular morpheme when it has already learned the plural morpheme than when it has not. It should also be easier for this network to learn the related rules for the past tense. We are currently testing the network on these problems.

---

[3] As far as we know, this is the first attempt to train a single network to perform both production and perception tasks.

We also need to be concerned about the plausibility of the approach to learning embodied in the SRNs used here. The prediction and auto-association tasks are essentially unsupervised, but the approach used in the second experiment assumes that the learner has available the correct meaning as well as the form of a word. A more plausible view is that on encountering a new word, the learner may have access to a number of semantic elements present in the context, including some which the speaker intended to convey through the use of the word. However, the hearer cannot normally distinguish relevant from irrelevant elements, nor is she often likely to have access to all of the relevant elements. We are currently attempting to implement these more reasonable assumptions.

An even more serious discrepancy between the experimental situation and real language acquisition is the fact that words are presented here in isolation. We plan to train SRNs on the task of Experiments 1 without the benefit of word boundaries. Our hope is that as the network begins to detect phonological regularities as well as frequently recurring sequences, it can use this information to guess at the locations of word boundaries, even for unfamiliar words. In fact, Kaye (1989) has argued that one of the primary functions of phonology in natural language is to aid in the segmentation process during parsing.

A further question to be asked of the present approach is whether it incorporates constraints on what is possible in human language. An often mentioned example is the non-occurrence of processes which reverse the segments in a morpheme, processes which the Rumelhart and McClelland (1986) model was quite capable of learning (Pinker & Prince, 1988). It is not immediately clear how difficult an SRN would find reversal, and this is a further ripe area for investigation.

Finally, we do not want to claim that this simple network architecture will be adequate for the acquisition of the phonological system of a natural language. We are nowhere close to being able to deal with data as complex as those covered by Lakoff's (1989a, b) and Touretzky and Wheeler's (1989) model. What seems certain is that a phonology acquisition system will need to have the capacity to develop abstract levels of knowledge above the segment. These levels, often called "tiers", have become central in phonological research since the work of Goldsmith (1976) and others arguing for a "non-linear" phonology. Various analyses posit separate levels for syllables, tones (in tone languages), multi-syllabic units, and timing units. In particular, current research emphasizes the significance of a syllable level. We think, however, that the representations that appear on the hidden layer in our networks are a step in this direction. That is, the pattern of activation on the hidden layer at the end of the presentation of a syllable is clearly a kind of distributed representation of that syllable. One question is how these representations might be used as inputs to another networks, ones designed to discover and represent patterns at a higher level of abstraction. This is the major focus of our future work.

8. References

Chomsky, N., & Halle, M. (1968). *The sound pattern of English*. New York: Harper & Row.

Elman, J. L. (1988). *Finding structure in time* (Technical Report 8801). La Jolla, CA: University of California, San Diego, Center for Research in Language.

Elman, J. L. (1989). *Representation and structure in connectionist models* (Technical Report 8903). La Jolla, CA: University of California, San Diego, Center for Research in Language.

Fodor, J. A., & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition, 28*, 3-71.

Goldsmith, J. (1976). *Autosegmental phonology*. Unpublished doctoral dissertation, Massachusetts Institute of Technology.

Jordan, M. I. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, 531-546.

Kaye, J. (1989). *Phonology: a cognitive view*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Lakoff, G. (1989a). A suggestion for a linguistics with connectionist foundations. In D. Touretzky, G. Hinton, & T. Sejnowski (Eds.), *Proceedings of the 1988 Connectionist Models Summer School* (pp. 301-314). Palo Alto, CA: Morgan Kauffman.

Lakoff, G. (1989b). Cognitive phonology. Paper presented at the Annual Meeting of the Linguistics Society of America, December, 1988.

Pinker, S., & Prince, A. (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition, 28*, 73-193.

Plunkett, K., & Marchman, V. (1989). *Pattern association in a back propagation network: Implications for child language acquisition*. (Technical Report 8902). La Jolla, CA: University of California, San Diego, Center for Research in Language.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel Distributed Processing: Explorations in the microstructures of cognition: Vol.1: Foundations* (pp. 319-362). Cambridge, MA: MIT Press.

Rumelhart, D. E., & McClelland, J. L. (1986). On learning the past tenses of English verbs. In J. L. McClelland, D. E. Rumelhart, & the PDP Research Group (Eds.), *Parallel Distributed Processing. Explorations in the microstructures of cognition: Vol. 2: Psychological and biological models* (pp. 216-271). Cambridge, MA: MIT Press.

Sejnowski, T. J., & Rosenberg, C. R. (1987). Parallel networks that learn to pronounce English text. *Complex Systems, 1,* 145-168.

Servan-Schreiber, D., Cleeremans, A., & McClelland, J. L. (1989). Learning sequential structure in simple recurrent networks. In D. S. Touretzky (Ed.), *Advances in neural information processing systems I* (pp. 643-652).

Touretzky, D. S. (1989). Towards a connectionist phonology: the "many maps" approach to sequence manipulation. *Proceedings of the 11th Annual Conference of the Cognitive Science Society,* 188-195.

Touretzky, D. S., & Wheeler, D. W. (1989). *A connectionist implementation of cognitive phonology* (Technical Report CMU-CS-89-144). Pittsburgh: Carnegie Mellon University, School of Computer Science.