Learning Words in Time: Towards a Modular Connectionist Account of the Acquisition of Receptive Morphology

Michael Gasser
Computer Science and Linguistics Departments
Indiana University
Bloomington, IN, USA
gasser@cs.indiana.edu

June 10, 1993

Abstract

To have learned the morphology of a natural language is to have the capacity both to recognize and to produce words consisting of novel combinations of familiar morphemes. Most recent work on the acquisition of morphology takes the perspective of production, but it is receptive morphology which comes first in the child. This paper presents a connectionist model of the acquisition of the capacity to recognize morphologically complex words. The model takes sequences of phonetic segments as inputs and maps them onto output units representing the meanings of lexical and grammatical morphemes. It consists of a simple recurrent network with separate hidden-layer modules for the tasks of recognizing the root and the grammatical morphemes of the input word. Experiments with artificial language stimuli demonstrate that the model generalizes to novel words for morphological rules of all but one of the major types found in natural languages and that a version of the network with unassigned hidden-layer modules can learn to assign them to the output recognition tasks in an efficient manner. I also argue that for rules involving reduplication, that is, the copying of portions of a root, the network requires separate recurrent subnetworks for sequences of larger units such as syllables. The network can learn to develop its own syllable representations which not only support the recognition of reduplication but also provide the basis for learning to produce, as well as recognize, morphologically complex words. The model makes many detailed predictions about the learning difficulty of particular morphological rules.

1 Motivation

By the time they are four years old, children understand and produce hundreds of words in the language(s) they are exposed to. What is more, they are well on their way to mastering the processes by which complex words are created out of the pieces that make them up. That is, they have begun to learn the **morphology** of the language(s).

The striking aspect of much of morphology is its **productivity**. Once children have learned how to form the regular past in English, for example, they can form the past tense of any regular verb, including those they have never heard before. The famous experiments of Berko (1958) showed

that preschool English-speaking children knew the rules for English past tense and plural. She demonstrated this by teaching children the present-tense forms of nonsense verbs such as *rick* and then providing a context in which they would produce the past-tense form. If children produced the appropriate form (i.e., *ricked*, with the *-ed* suffix realized as a /t/), this was evidence they knew the past-tense rule. Further evidence that children make generalizations about morphology is the well-known fact that they tend to over-generalize; that is, they regularize irregular forms. For example, in English they produce past-tense forms like *goed* and *taked*. Since these are forms they have probably never heard before, they are clearly applying the rule they have learned for regular verbs.

But what is the nature of the rules that are being learned? The conventional view is that the rules are **symbolic**. This means, in particular, the following.

1. Rules are all-or-none. A system either has a given rule, or it doesn't.
2. Rules are localized. Knowledge is not shared between rules.
3. Rules make reference to abstract morphological variables such as **root** and **affix**.
4. Rules implement the operations of concatenation, insertion, exchange, copying, and segment association.
5. Rules apply to representations at varying levels of abstraction. These representations have explicit **structure**.

Within this framework, any irregularities within the system are handled by a mechanism entirely separate from the rules, by a module which specializes in rote memorization. As a simple illustration, consider the regular English past tense rule. Underlying the various forms of a regular English verb is a **stem** form. To form the past tense, a suffix is catenated onto the end of the stem form. A morphophonological rule specifies the precise form that the suffix takes: /t/, /d/, or /ɨd/, depending on the last segment in the stem. As we will see below, there are much more complicated morphological processes than this, but the basic elements of a symbolic analysis are there. The rule described pertains to the suffixation of the various realizations of *-ed* on verb stems; it shares nothing with any other rules concerning English morphology. This is not to say that a more general rule is not possible. For example, there are other morphemes that take exactly the same form as the past tense affix (e.g., the past participle morpheme), and the plural morpheme shows a parallel variation in form. The important point is that, whatever form the rule takes, it applies to a clearly specified set of environments and no others; rules are not distributed. Furthermore, the rule makes reference to the stem of the verb. In the case of English, the verb stem happens to have the form of the present tense (other than the third person singular), but the stem in many other cases is not a form that actually occurs as a word in the language. This is the case, for example, with a Spanish verb. The verb *cantar* 'sing', for example, has the stem *cant*, to which various tense/aspect/person/number/gender suffixes are added, but this form occurs nowhere in the language as an independent word. Finally, the process of affixation consists in concatenating the stem and the suffix.

While there is of course disagreement about the details, a picture like the one I have sketched here was until recently accepted by nearly everyone as the only way to deal with the morphology of natural language. With the advent of connectionist models, however, this view has come under question. Connectionist models have none of the five properties listed above. "Rules" in

connectionist networks apply to varying degrees rather than in an all-or-none fashion, are distributed across network weights, do not make reference to variables, do not implement concatenation or other symbolic operations, and do not apply to representations with explicit structure. While this seems to make networks poorly suited for phenomena such as morphological rules, nearly everyone seems to agree that something like a connectionist pattern associator is required for "low-level" cognitive phenomena, including for example, the learning of irregular morphology (Kim, Pinker, Prince, & Prasada, 1991). For the sake of parsimony then, if nothing else, it is of interest to know how far connectionist networks can go in modeling the learning or morphological rules.

Connectionist morphology began with the well-known and controversial paper by Rumelhart and McClelland on the acquisition of the English past tense (Rumelhart & McClelland, 1986). This model consisted of a simple perceptron which was trained to associate English verb stems with past tense forms. The network was capable of learning both regular and irregular forms and, for many novel verbs, that is, stem-to-past-tense associations on which it had not been trained, generalizing to the correct past tense form. It also exhibited overgeneralization; that is, irregular verbs which the network had been trained on were treated as regulars. This overgeneralization is characteristic of children's speech.[1]

The Rumelhart and McClelland paper has generated perhaps more discussion and further research than any other paper dealing with connectionism and high-level cognition. Among the connectionist models directly addressing the English past tense are Cottrell & Plunkett (1991), Daugherty & Seidenberg (1992), Hare & Elman (1992), Hoeffner (1992), MacWhinney & Leinbach (1991), and Plunkett & Marchman (1991); other connectionist approaches to morphology include Corina (1991) and Gasser & Lee (1991).

I will not attempt to review in any detail the ensuing debate and the various descendants of the Rumelhart and McClelland model. Most of the controversy has concerned two questions. First, the means by which the phonology of input and output forms was represented in the Rumelhart and McClelland network left much to be desired, and there have been several alternative approaches suggested. The question of how the phonetic input to a network is to be represented will be of some concern in this paper. Second, there has been a great deal of discussion concerning whether a single mechanism, in particular, a pattern associator of some type, could in fact accommodate what is known about both regular and irregular morphology (Kim et al., 1991; Marcus, Brinkmann, Clahsen, Wiese, Woest, & Pinker, 1993; Pinker & Prince, 1988). I will have little to say about this debate in this paper,

Rather than be concerned with irregular morphology and overregularization, I will be interested in a wide range of **regular** morphological processes. If connectionist networks are to be taken seriously as models of how natural language morphology is acquired and processed, it will not suffice to show that a simple suffix rule like the English past tense can be learned. Thus the first goal of this paper, as in related earlier work (Gasser & Lee, 1991; Lee, 1991), will be to investigate how far relatively simple sorts of networks can go in handling a variety of morphological rules and in what sorts of motivated augmentations help where they seem to fail.

A second concern is with the way the task of learning morphology is defined in the first place. Most models, whether connectionist or symbolic, assume that what is learned is a mapping from one form to another, where one of these may be a relatively abstract internal representation of some type. But for children the immediate task is that of understanding the words they hear and producing

[1]There is disagreement, however, on how frequent and long-lasting a phenomenon it is (Marcus, Pinker, Ullman, Hollander, Rosen, & Xu, 1992).

words which others will understand. That is, they must learn to map forms onto meanings and meanings onto forms. A small number of models have taken this approach (Cottrell & Plunkett, 1991; Gasser & Lee, 1991; Hoeffner, 1992), and the authors in each case have noted other benefits of posing the problem in these terms. In particular, the inclusion of semantics makes it possible to represent homonymous words, and the fact that the semantic input for one tense resembles the semantic input for the other causes the rule which makes no change at all to be the default rule, as it should be. However, these models, with the exception of Gasser & Lee (1991), the predecessor of the model described here, treat word production only. The focus on production apparently results from an interest in modeling results like those of (Berko, 1958), which are production-oriented tests of a knowledge of rules. But children learn to understand words before they can produce them, and, as argued in more detail below, the production capacity must build on the comprehension capacity. The approach argued for here gives primacy to comprehension, that is, word recognition, and attempts to explain how comprehension and production are related. Finally, most models of the processing or acquisition of morphology ignore the fact that words take place in time. But this precludes an examination of any effects that the temporal nature of language might have on the learner/processor, and in connectionist models, it makes the representation of words of varying lengths awkward. I will show how the sequential nature of words can be accommodated in a model of the acquisition of receptive morphology and how treating words in this way leads to predictions about the relative difficulty of morphological rules.

This paper is concerned, then, with the question of whether a connectionist network can learn to recognize morphologically complex words, presented one segment at a time, and created from a variety of morphological processes. The problem is one of segmentation in time: given input sequences and information about what parts the sequences consist of, though not where the parts are, the network must learn how to identify the parts in unfamiliar sequences.[2] The extent to which it succeeds will depend on properties of the rules and of the phonology of the target language, and an important goal of this project is a teasing apart of the factors that make some rules more difficult than others for a network. The result should be predictions about relative difficulty, many related to the sequential nature of the network's input, predictions which later be verified experimentally. Of particular interest will be the differing effects of particular factors on the recognition of lexical morphemes (such as *play*) and grammatical morphemes (such as PAST TENSE). In addition, because these two subtasks make very different demands on a network, I will show that the performance of a network is improved dramatically when it is augmented with modules assigned to the two subtasks. I also argue that for modeling the learning of one category of morphological rule, reduplication, a further type of modularity seems to be required, one which separates the subnetwork responsible for sequences of phonetic segments from that responsible for larger units such as syllables.

Handling a complex phenomenon such as morphology has two aspects. First, the hypothesized mechanism must be powerful enough to deal with (in this case, to learn) the range of processes that occur. Second, the mechanism must not be too powerful; whatever constraints there are on the phenomenon being modeled should fall out naturally in the model. That is, the model should fail to handle those sorts of processes which are impossible for people to learn. A minor goal of the paper is to illustrate for one example how the model fails on one type of process which appears not to occur in the morphology of natural languages.

The remainder of the paper is organized as follows. First, I give an overview of what is seems

[2]The question of how a network might learn to **produce** novel words and how this ability is based on the ability to recognize words is the topic of a second paper.

to be possible in the the morphology of natural languages. Second, I discuss morphology from the perspective of the child learning the language and propose a set of desirable features for a model of the acquisition of receptive morphology. Next, I present the model itself. A series of experiments is described which demonstrate that a recurrent network, augmented with two sorts of modularity, generalizes on all of the basic types of morphological rules. I examine in some detail the nature of one of these types of modularity and show that a network can learn to make use of the modules provided to it in an efficient way. Finally, I discuss the implications of the model for factors affecting the difficulty of morphological rules, for the status of modularity in networks and in natural language, and for the role of time in natural language processing.

2 Natural Language Morphology: Some Basic Issues

In this section, I give a highly simplified introduction to the morphology of natural languages. For a more comprehensive treatment, see (Spencer, 1991). The approach adopted here will be the traditional linguistic one: I will be looking at the phenomena from the perspective of the language as a system, rather than from the perspective of a human (or machine) user or learner.

Morphology is about the way in which meaningful pieces, **morphemes**, are put together to make words. Languages differ greatly in terms of how much of the work of expressing meanings is handled by morphology. On the one extreme are languages such as Vietnamese, where the only words consisting of more than one morpheme are compounds, made up, for example, of a verb plus a noun. On the other extreme are languages such as the members of the Eskimo family, in which a single word may contain a dozen morphemes and correspond to an entire sentence in a language such as English. In this paper, I will be concerned only with morphological processes which form words consisting of a single **root** conveying some lexical content and one or more **grammatical morphemes** representing **grammatical categories** that modify the meaning of the root. Each grammatical category, for example, tense, represents an exhaustive set of meanings of a particular type (e.g., PRESENT, PAST, FUTURE) and a corresponding set of morphemes, only one of which appears on a given word. As an example, consider the Swahili verb *nilikuona*, which means 'I saw you (singular)'. This word consists of a root and four grammatical morphemes, each representing a different morphological category. The root is the morpheme *-on-*, meaning 'see'. The four grammatical categories are subject person/number, marked by the prefix *ni-*, 'subject = I'; tense, marked by the infix *-li-* 'past'; direct object person/number, marked by the infix *-ku-* 'direct object = you (singular)'; and mood, marked by the suffix *-a* 'neutral (not negative, not subjunctive) mood'.

Grammatical morphemes are often divided into **inflections** and **derivational** morphemes, but the distinction will not be important for our purposes. For convenience, I will in general refer to grammatical morphemes as "inflections". I will also refer to the form to which the inflection is added (when it is added) as the **stem** of the word. As we will see below, the stem of a word is not necessarily the same as the root of a word.

2.1 Morphology and Meaning

Clearly morphology is located at that point where form and meaning meet. Work on semantics (and syntax) from the perspective of morphology focuses on the functions that grammatical morphemes

can have. One could argue that there is a fundamental distinction between the meanings of grammatical morphemes (notions such as PLURAL, FIRST-PERSON, and COMPARATIVE) and the meanings of lexical morphemes (notions such as DOG, BREAK, and OLD). In particular, grammatical meanings are normally organized in sets with a small, fixed number of members, what I have referred to above as grammatical categories, e.g., number, tense, and definiteness, while there seems to be no limit on the new lexical meanings that can be created. The conventional view seems to be that it is not necessary to have an understanding of what notions such as PRESENT and BREAK really are in order to understand how morphology works, and I will maintain this position here. On this view, the meanings of morphemes are unanalyzed primitives, organized in terms of whether they are lexical or grammatical, and, if they are grammatical, which category they belong to.

With respect to lexical morphemes, in most accounts there is a further elaboration: mediating form and meaning is a separate level of **lexical entry**. By these accounts it is the lexical entry rather than the semantics, for example, which is associated with an irregular form. This is deemed necessary to explain the fact that speakers treat polysemous words in a unitary fashion morphologically: the past tense of *get* is *got* no matter what it means (Kim et al., 1991).

2.2 Types of Morphological Processes

What sorts of possibilities are there for the ways in which roots and inflections combine to form words? The most common type is **affixation**, by which the inflections (the **affixes**) are "attached" to the root (or, more precisely, the stem). There are four possibilities. Most commonly, the affixes are concatenated onto the front (prefixation) or the end (suffixation) of the root. Examples from English: *un-tie*, *small-er*.

Another possibility is for the affix to be composed of portions appearing at both the front and the end of the root (circumfixation).[3] A German example: *ge-mach-t* 'done' (root: *mach-* 'do'). Finally, the affix may be inserted within the root (infixation). An example from Tagalog, a Malayo-Polynesian language of the Philippines: *s-um-ulat* 'to write (subject focus)', *s-in-ulat* 'to write (direct object focus)' (root: *sulat* 'write'). Infixation is clearly more complex than either prefixation or suffixation because of the need to specify where the infix is to appear. For different infixation processes, this may require reference to syllables or other units larger than the individual segment within the root. Infixes may also differ in whether their position within the stem is specified in terms of the beginning of the stem (e.g., "following the first vowel") or the end of the stem (e.g., "preceding the penultimate syllable").

A process which may be considered a form of affixation is **reduplication**, which consists in the addition of a copied portion of some part of the stem. There are various possibilities for the source of the copy and the target, that is, where the copied portion ends up, but there also appear to be severe constraints. In any case, a full treatment of the possibilities is beyond the scope of this paper. An example from Madurese, a Malayo-Polynesian language spoken in Indonesia: *buwaqan* 'fruit', *waq-buwaqan* 'fruits'. Here the sequence *waq* in the singular form is copied onto the front of the root to form the plural (Stevens, 1968). The fact that a portion of the root is copied makes reduplication considerably more complex than affixation. The statement of a reduplication rule seems to require a variable of a sort not necessary for an affixation rule.

It is also possible, though very rare, for material to be **deleted** in a morphological process. In

[3] Some would argue that such cases involve separate prefixation and suffixation processes. The arguments need not concern us here.

Koasati, for example, verbs have plural forms which are formed by deleting a portion of the singular forms: *lasaplin* 'lick (singular)', *laslin* 'lick (plural)' (Martin, 1988).

A very different sort of process, which I shall refer to as **mutation**, consists in an alternation in one or more of the root segments. This is familiar in the formation of the past tense of irregular ("strong") verbs in English and other Germanic languages (e.g., *swim*, *swam*, *swum*), but it is a completely regular process in many other languages. An example from Chichewa, a Bantu language spoken in Malawi: *ndímafotokózá* 'I explain (habitual)', *ndímafótókoza* 'I explained (habitual)', The accent mark marks a syllable spoken with a relatively high tone. Thus in this languages changes in tone alone can indicate a difference in tense (Mtenje, 1987).

One final possibility may be considered either a form of affixation or a form of mutation, but it deserves special mention because of its complexity. It is best known for verbs in the Semitic languages. In these languages, all that the various forms of a verb share is a set of consonants, usually three, which always appear in the same order but are separated by different other segments, usually vowels. Each grammatical category specifies a template which provides the intervening segments and in some cases stipulates that consonants are to be doubled ("geminated"). An example from Amharic, a Semitic language spoken in Ethiopia: *mä-sbär* 'to break', *säbbär-ä* 'he broke'. The root of this verb is the consonant sequence *sbr*, and the two forms (infinitive and past tense) fill in the intervening vowels and, for the past tense, double the second root consonant. This sort of morphological process may be referred to as **templatic**. Note that these forms also illustrate prefixation (the *mä-* of the infinitive) and suffixation (the *-ä* of the third person singular past tense).

In summary, morphological processes can (1) add material, either before, after, or within the root, (2) delete material (though very infrequently), (3) copy portions of the root, usually to the position either before or after the root, (4) alter segments within the root, and (5) specify a template which intercalates segments between the root segments. As already seen in the Swahili and Amharic examples, roots may combine with more than one grammatical morpheme to form words. In such cases more than one type of morphological process is often involved, for example, a templatic process and suffixation in the case of the Amharic verb *säbbär-ä*.

A key notion in standard accounts of morphology and phonology is that of an **underlying representation** for a word. The underlying representation is an abstract characterization of the form of a word in which each morpheme is expressed in a context-independent fashion, that is, independent of the other morphemes making up the word. It is abstract in the sense that it may correspond to a form which does not occur on the surface. The **derivation** of the word is the process that takes the underlying representation and yields a surface representation of the word. For example, for the derivation of the Spanish verb *hablo* 'I speak', we have something like *habl-* + *-o* $\rightarrow$ *hablo*, where *habl-* is the root (or stem) of the verb 'speak' and *-o* marks the first person singular present. To take a more complicated example, consider the Amharic verb *säbbär-ä*. Here the derivation would be something like the following: *sbr* + C_1*ä*C_2C_2*ä*C_3 + *ä* $\rightarrow$ *säbbär-ä*. Here the rules of the derivation, among other things, specify how the root consonants are assigned to the consonant positions (C_n) in the past tense template. Note how this Amharic example provides motivation for the notion of an abstract underlying representation. A characterization of where the various surface forms of a verb come from seems to require a level at which there are representations of forms which never occur on the surface (the root consonant sequence) or are expressed in terms of abstractions such as C_2.

2.3 Morphological Constraints

Morphology is constrained in three sorts of ways. First, within particular languages there are constraints on the environments in which particular morphemes can occur and on the ordering of morphemes. Thus in a Swahili verb such as *nilikuona* (see above), the subject prefix must precede the tense infix, which must in turn precede the object infix, if there is one.

Second, the phonology of the language may be constrained in ways which affect the morphology. If a language does not permit sequences of vowels, a vowel suffix attached to a stem ending in a vowel would lead to an illegal form. Under these circumstances, we often see alternation in the shape of the morpheme. In this case, for example, a consonant might be inserted between the vowels to break up the sequence. Such processes are referred to as **morphophonology**. Another example of a morphophonological alternation occurs in languages with what is referred to as **harmony**, that is, constraints on what sorts of segments can occur together in a word. If a word can contain only either rounded or unrounded vowels, for example, there may be two forms for an affix, one with a rounded vowel for stems containing rounded vowels and one with an unrounded vowel for stems containing unrounded vowels.

Third, there appear to be universal constraints on what sorts of morphological rules are possible. I will discuss only one proposed constraint here. To understand this constraint, we will need to look a little more closely at the type of analysis that has come to dominate phonology, and to some extent, morphology in recent years. This approach, known as **auto-segmental** phonology/morphology, posits a set of relatively independent **tiers** within which aspects of a word's form are represented at the underlying level. This is clearest for languages such as Amharic which seem to require abstract representations. For this language, one possible analysis would place the root, the tense template, and the person/number/gender affixes on three separate tiers. That is, there is a tier for each morpheme in a verb. The derivational process can then be seen as a matter of associating the positions in each tier with positions on a **skeletal tier** representing the surface locations of consonants and vowels. The left side of Figure 1 shows the associations for the Amharic verb stem *säbbär*. Within languages like Amharic, there are many variations on this scheme. There seem not to be any languages, however, in which the lines that associate segments on different tiers cross each other.[4] This is what would happen for the imaginary Amharic form shown on the right side of Figure 1, *säbräb*. Another way of stating the constraint is to say that the relative position of two segments within a morpheme remains the same in the different forms of the word. One of the arguments in favor of autosegmental analyses, in fact, is the ease with which the constraint can be stated: association lines do not cross.

I will return to this constraint later in the context of the model being proposed in this paper.

3 Processing and Learning Morphology

Our concern in this paper is with the *use* and *acquisition* of morphology. That is, we will be interested in whatever processes get a hearer from a "surface" form to the set of meanings that the word is meant to convey and how a child manages to learn these processes. In this section I discuss

[4]This use of distributional evidence has been criticized on the grounds that the absence of a feature from the world's languages may be a historical accident rather than a reflection of some deep-seated processing or learning mechanism (Pullum, 1982). I will only assume that this absence might lead one to posit the feature as a constraint, but further evidence would be required to confirm this hypothesis.

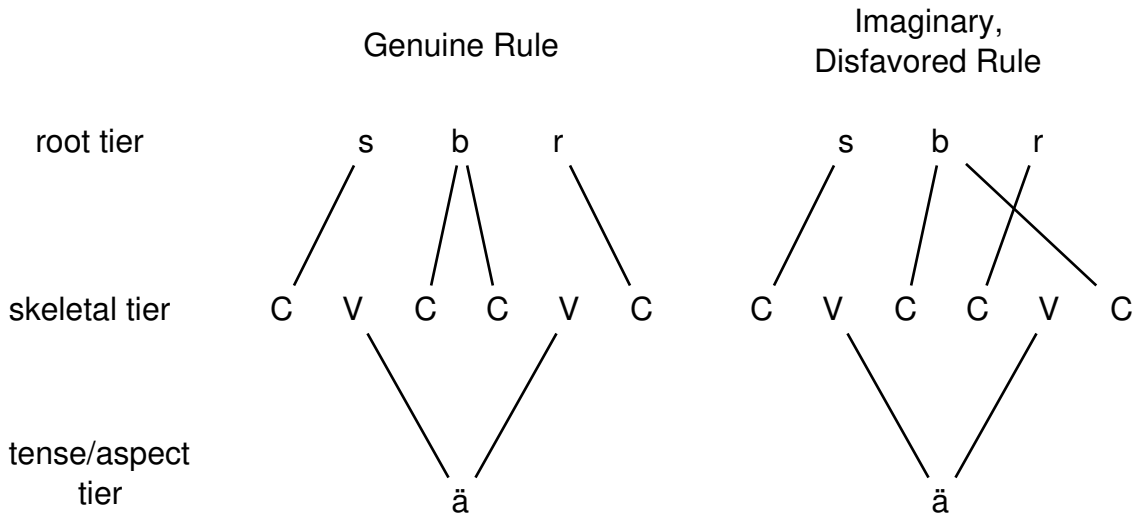

Figure 1: Auto-segmental representation of Amharic verb stems, one genuine, one imaginary and disfavored

what considerations in addition to the linguistic ones outlined in the last section are necessary for a characterization of these processes. These considerations concern the nature of the inputs and outputs to learning and processing and the way in which in the learner is guided by the environment in the learning process.

3.1 Form and Meaning

We must first consider the question of what a surface form is. What a hearer has initial access to is a wave form, but the processes involved in extracting phoneme-like segments or larger units from wave forms constitute in and of themselves an entire field of study. To simplify matters (and possibly muddle them), I will assume that the input to the word recognition mechanism and the output of the word production mechanism is a sequence of phonetic segments. Though somewhat dangerous, this is not an unconventional assumption. The sequential nature of the input is discussed in some depth in the next section.

At the other end of word recognition is semantics. Whether or not the process is mediated by a separate level of lexical entries is a question I will not address in this paper. For the purposes of this paper, I will be assuming that the output of recognition consists of a set of unanalyzed primitives, one for each morpheme. Thus, because non-arbitrary relationships between form and lexical meaning will be of no concern here, each lexical morpheme is treated as a single unit at the output rather than treated as a set of semantic features, as it is, for example, in the production model of Cottrell & Plunkett (1991). I will refer to the output of word recognition, defined in this way, as "semantics", though it can be seen at best as a pointer to the real semantics of the word being recognized. The fact that lexical morphemes are localized means that the lexical outputs could just as well be treated as lexical entries, rather than as semantics. Nothing here will hinge on the distinction between the two.

It is worth noting that the semantics of words on this account is simplified in a further way since

the hierarchical relationships among the meanings of the morphemes are ignored; the meaning of a word is a *set*, not a *tree*.

3.2 Time and Short-Term Memory

Language takes place in time. Words and sentences become available to listeners a little at a time rather than all at once, and words and sentences are uttered by talkers a little at a time. This would not matter much if it were not for the fact that language is organized in terms of units of various sizes. Thus perception apparently involves the segmentation of an input stream into words, phrases, and probably other units at other levels. But in order to recognize a unit such as a word, the language processing system requires access to a whole stretch of input, more, that is, than it has direct access to at any given time. What this means is that the perception of language requires some form of **short-term memory**. Production also requires a short-term memory because, in order to know what to produce next, the system must remember what portion of the current unit, say, word, it has already produced.

Because language takes place in time and linguistic forms are composed of units, language processing calls for a short-term memory. This much is hardly controversial. The important question for our purposes is how "high-level" an issue time is. If, for example, relatively low-level processes turn the temporal pattern representing a word into a static pattern, which is then recognized as being one word or another, then we can relegate temporal processing and short-term memory to acoustic/auditory phonetics and ignore them here. If, when a word is produced, a phonological representation for the entire word becomes available at once, and the translation of this pattern into a sequence of articulatory gestures is accomplished by low-level phonetic processes, then we are again justified in ignoring it here. If, on the other hand, word recognition and word production are **incremental** processes, the temporal nature of words and the nature of linguistic short-term memory are of concern to us. In fact, there is considerable psycholinguistic evidence that word recognition and production are incremental. Words are often recognized long before they finish; hearers seem to be continuously comparing the contents of a linguistic short-term memory with the phonological representations in their mental lexicons (Marslen-Wilson & Tyler, 1980). And production of a word, or some portion of a word, is often initiated before the word has been completely formulated (Kempen & Hoenkamp, 1987). Thus it is simply not the case that entire words (or other units) are available as static chunks for word recognition or production.[5]

Linguistic forms, then, are temporal objects. Since we are not dealing here with continuous input to the perceptual system, we can think of linguistic forms as **sequences**. But semantics, that is, the output of word recognition and the input to word production, is apparently not sequential. Or, if it is, its time course is unrelated to the time course of the words associated with it. For recognition, once a form is interpreted, we can safely assume that the semantic output remains constant until the input word is complete. And for production, it is reasonable to assume that semantic input remains constant throughout the production of a word. Thus the processing of words is the mapping of sequences of phonetic segments onto static semantic patterns.

In sum, the processing of words requires a short-term memory of some sort to store previous context. From the perspective of time, perception consists in mapping sequences at a lower level

[5]Entire words in a skeletal form *are* apparently available to production, but the spelling out of the skeleton is an incremental process above the level of articulatory gestures. This is a feature of several models of production, e.g., Dell (1986). How it fits into the present model is discussed in a later paper.

onto static representations at a higher level, and production consists in the reverse process. The picture thus far is illustrated in Figure 2.

In the discussion of the model, I will return to the issue of time and short-term memory in the context of how best to implement a short-term memory in a network.

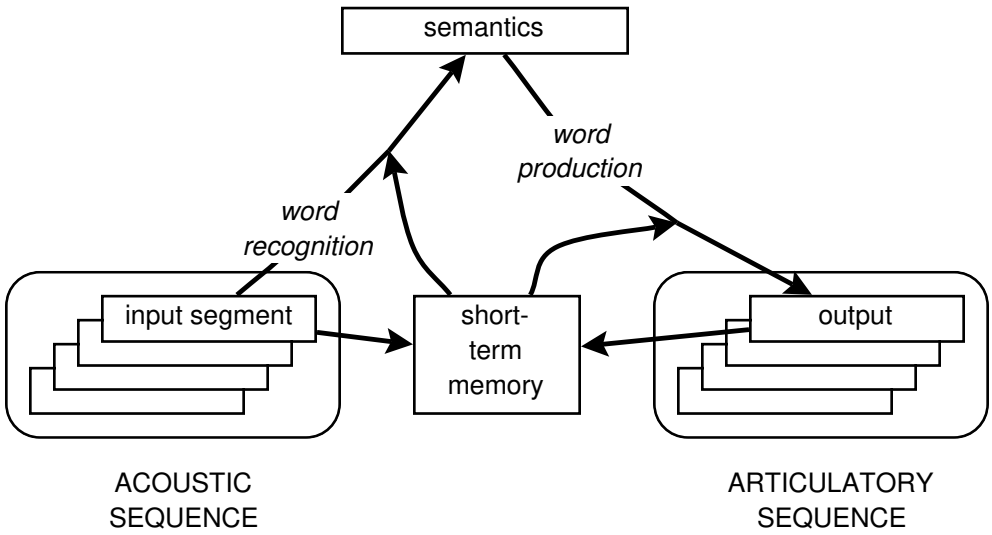

Figure 2: Overview of word recognition and production

3.3 Learning to Recognize Words

Now consider the problem of *learning* to recognize and produce words, as these processes were defined in the last section.[6] Suppose a child is faced with a novel monomorphemic word, say, the word *frog* in the sentence *look at the frog over there*. If the remainder of the sentence is familiar, the child should be able to identify the boundaries of the unfamiliar word. In order to make sense of the word, the child makes use of the fact that the meaning can often be inferred from context. In this example, there would presumably be a relatively salient frog present. In the case of polymorphemic words, the child would again, at least on some occasions, have access to some or all of the semantic features signalled by the word. Thus in the case of the word *frogs* in the sentence *look at the frogs over there*, there would be a salient set of frogs present; both plurality and frogness are available from the environment.

Thus, to an approximation, the task of learning to recognize words is one in which the learner is presented with a set of pairings of input forms and target semantic representations. Because an explicit target is available, the learning of word recognition can be seen as an instance of **supervised learning** (Hertz, Krogh, & Palmer, 1991).

Now, following some period of such learning, when the child is faced with a novel polymorphemic word—one representing an unfamiliar combination of familiar morphemes—she may be able to interpret the word by using the morphological rule that she has learned.

[6]For a more thorough discussion of many of the issues involved in the acquisition of morphology, see MacWhinney (1978)

3.4 Learning to Produce Words

Though it will not be the major focus of this paper, I will also consider what might be involved in learning to produce words since this will have a bearing on how the learning of receptive morphology takes place.

Suppose the child is faced with the task of producing a novel word. In the case of a monomorphemic word, this means that the child must produce a morpheme she has never produced before. This would be analogous to the recognition of a novel monomorphemic word if the appropriate target were available. While the environment often provides the semantics for an unknown form, the environment generally does not provide the appropriate form for an intended meaning. A child who wants to say *look at the frog* but who does not know the word for FROG normally gets no help from her surroundings. Under certain circumstances, of course, the correct form is made available. She may make an explicit request for the word (*what do you call that?*), or a helpful interlocutor may guess what she is after and provide the word. Alternatively, the child may produce the wrong form (*look at the snake*) and get corrected, assuming the context makes it clear what she intended. More often than not, however, an intent to convey a meaning for which the child does not know the word results in no learning at all because no target is provided. This state of affairs is even more likely in the case of polymorphemic words. Say the child knows the word *frog* and wants to refer to a set of frogs but does not know how to make the word plural. One likely possibility is for her to go ahead and use the only form of the word she knows: *look at the frog*.[7] She may be corrected under these circumstances, though correction is less likely here than in the monomorphemic case because, from the perspective of getting the intended point across, the error is not so serious. And even if the error is corrected, the child may choose to ignore the correction. All this is assuming the incorrect form that the child utters is comprehensible. When it is not, then there is of course no possibility of correction.

Thus while there are certainly some cases in which there is an explicit target for the production of an unfamiliar word, these will be the exception rather than the rule. How then do children learn to produce words? There are two logical possibilities. Either they learn by attempting to analyze their own output, or they learn production as they are learning recognition.

By the first possibility, children treat the output of their own production system as input to their word recognition system. If they have learned the component morphemes and whatever morphological and phonological rules apply in the recognition direction, they are in a position to evaluate their production output by comparing the semantic output of the recognition process with the original intent behind their production. This mechanism is a part of the symbolic model of morphological acquisition described in MacWhinney (1978). In this case, however, there is still no target for production. What is available rather is some measure of how correct the output was. This would thus be an instance of **reinforcement learning** (Hertz et al., 1991), which is considerably less powerful than the supervised learning which characterizes word recognition. This is due to the **credit assignment problem**: when the output is wrong, there is no direct way of knowing what led to the error.

The second way in which word production might take place is as a kind of side-effect of the learning of recognition. Since the learning of recognition can be viewed as a pairing of form with meaning, the reverse meaning-to-form mapping could be acquired, or strengthened, simultaneously

[7] An issue completely ignored here is that of how the child *decides* to refer to plurality in the first place. In many languages the marking of plural is either not obligatory, as it is in English, or not even possible.

with the form-to-meaning mapping. This would constitute genuine supervised learning since there would be a target for the production direction. However, this is only possible if a form *that is suitable for production* is made available during perception. What is labeled "acoustic sequence" in Figure 2 would not provide a suitable target for production. The clear implication is that, for this sort of learning to succeed, recognition and production must share representations. While there are a number of possibilities for the level of abstractness at which the sharing takes place, the representations must be form-, rather than meaning-based; otherwise the learning of the forms of words for production could not be based on the learning of form for perception. Figure 3 shows how the existence of **intermediate phonological representations** (IPRs) shared by recognition and production make possible some production learning during perception. This learning would take place between the areas labels "semantics" and "IPR".

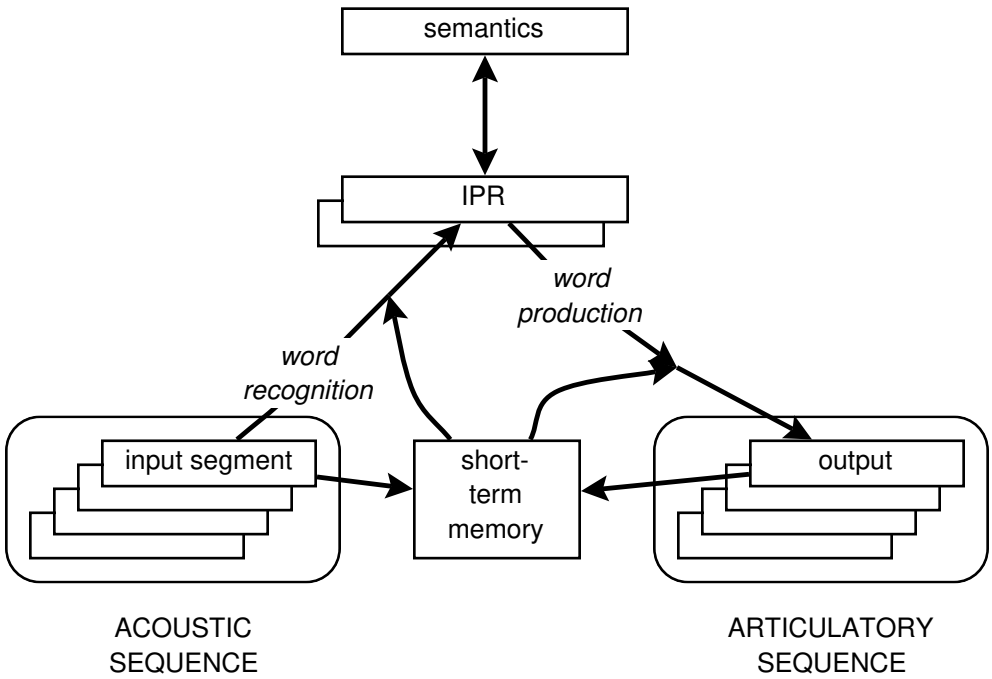

Figure 3: Shared representations for word recognition and production

In sum, production is only learnable if it is based on IPRs which are shared with perception. Since recognition invariably precedes production, at least in first language acquisition, it is reasonable to imagine that IPRs develop primarily out of the recognition process.

This does not solve the problem of how IPRs come to be shared. What it does do is permit the learning of production to be broken into two more or less discrete phases (which may overlap). In one phase, children learn to map audition/acoustics onto articulation. The result of this purely phonological learning, which may be largely a result of babbling, is shared IPRs. During the second phase, children begin to learn the mapping from form to meaning and from meaning to form. As they learn to recognize words, they also learn to represent them in terms of the IPRs developed during the first phase. Learning to produce words is then a matter of learning the mapping from meanings to the IPRs for words that arise during recognition. During this phase, there should also be further development of the IPRs themselves because of the phonological distinctions that only

become evident when words are learned.

There is nothing especially controversial in suggesting that there are phonological representations which are shared by perception and production. If there were not, all of the work of phonology would be irrelevant to models of the actual processing of language. There are, in any case, reasons related to efficiency which lead one to believe in the psychological reality of IPRs. Given the overwhelming amount of information present in the acoustic signal, it is unlikely that it is stored as such for anything but the shortest intervals. Rather more abstract representations which factor out what is irrelevant in the signal are called for. From the perspective of production, similar considerations apply. Since the system must store the forms of tens of thousands of words, an efficient means of storage is called for, one which capitalizes wherever possible on redundancy and regularity in the forms. Language is characterized by a great deal of such regularity—it is precisely this that phonologists have elucidated—so it would be surprising if the system did not make use of it in representing word forms internally.

It will however be useful to make a distinction between, on the one hand, IPRs as posited here and as realized in the model to be described and, on the other, the underlying representations of phonological research. Linguistic phonological representations are the result a particular type of linguistic analysis. While linguists may be reasonably systematic in deciding on representations for morphemes, I am unaware of an algorithmic statement of how they are arrived at given a set of data from a language. IPRs, on the other hand, are intended to emerge from a system that learns and processes language, that is, a child. They are a part of the solution of the phonology and morphology acquisition problem, not an input to a phonology or morphology processor. The challenge is to specify what sort of device has the capacity to evolve the IPRs that can mediate word recognition and production. The model presented here will offer a partial response to this challenge.

4 Summary

I have identified the following features as desirable in a model of the acquisition of morphology:

1. The model should learn mappings between forms and meanings.
2. The model should learn to recognize and produce words embodying rules of all of the types that occur in human languages.
3. The model should recognize and produce words involving combinations of rules.
4. The model should learn, again for both recognition and production, the sorts of phonological rules which are conditioned by morphological processes.
5. The model should embody the constraints and tendencies that characterize morphology and phonology.
6. The model should have a contextual short-term memory and should explain how sequential patterns map onto static patterns in recognition and vice versa in production.
7. Production in the model should be based on phonological representations which are learned for recognition.

This paper describes a model which addresses to one degree or another all of these issues for recognition.

5 Overview of the Model

5.1 The Task

The task of recognizing words, for the purposes of this paper, consists in mapping sequences of phonetic segments onto localized representations of one or more morphemes which make up the word, the latter provided as targets by the environment. I will refer to the output as "semantic", though, as noted above, it has no internal semantic structure. Production builds on representations learned during recognition; details of production will be the focus of a later paper.

5.2 The Problem of Time and Short-Term Memory

Word recognition and production, as defined here, might be modeled as simple pattern association tasks if it were not for the fact that they take place in time. Because what has already happened is relevant to the processes, there is a need for a short-term memory. Treating words as sequences rather than static inputs/outputs also greatly simplifies the representation of words of varying length and, since the same units are responsible for each segment in the word, permits generalization to be made across the segments.

Feedforward connectionist networks can be outfitted with a short-term memory capacity through the use of time delays on connections from inputs (e.g., Waibel, 1989). For example, if in addition to the ordinary connections from input units to hidden or output units, there are additional sets of input connections with delays of one and two time steps, then the system always has access to a window of width 3. The main disadvantage of such a short-term memory is its inflexibility. A problem for which a context longer than three items is required is unsolvable.

The simplest alternatives to sliding-window memories of this sort are networks in which the past context is in effect compressed into a pattern of fixed width, with no fixed limit on the length of the interval that is recorded. The past context may be either a compressed record of the network's output, as in the architecture due to Jordan (1986), or its input, as in the architecture due to Elman (1990), generally known as a **simple recurrent network**. For recognition, the output of the network should approach a fixed target, the semantics for the correct set of morphemes, as more and more of the word becomes available. Early in the word, the network's output should reflect uncertainty about the semantics because many words may be consistent with the input seen so far. Therefore, a short-term memory which records the network's output should not be particularly informative. It is rather the network's input that needs to be remembered. For this reason in the present model, a version of the simple recurrent network is applied to the task of recognizing words.[8]

A simple recurrent network achieves a compressed record of the hidden layer through the use of trainable recurrent time-delay connections on some or all of the network's hidden units. Such a network is illustrated in Figure 4. The arrows indicate complete connectivity between layers of units, and the sequential nature of the network is indicated by the overlapping boxes, each corresponding to the activation for a layer at a particular time step within a sequence. Each hidden unit has a connection to every other hidden unit with a time delay of one time step. On any given time step, this gives the hidden layer access not only to the current input but also to whatever pattern appeared on the hidden layer in the last time step. Since the previous hidden layer pattern also depended on the hidden layer for the time step before that, the network has access to points arbitrarily far back

[8] For a fuller discussion of issues related to the processing of temporal processes in networks, see Port (1990).

in time. Note that in this scheme the network can learn not only to use the short-term memory (through the training of the connections from the hidden to output layer) but also to differentially weight different points in the past or different aspects of the input by the training of the recurrent connections on the hidden layer. The usual way to implement an Elman network is by copying the hidden-layer to a **context** layer of the same size. The pattern on the context layer is input on each time step together with the external input to the network.

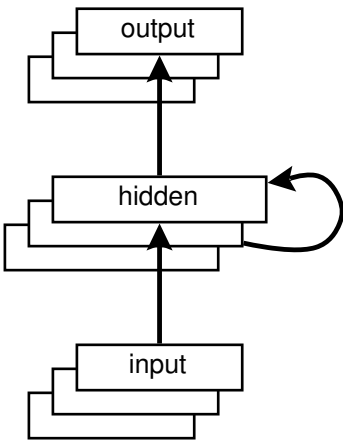

Figure 4: Elman network

There remains the problem of how to train simple recurrent networks. Simple moving-window networks, which are feedforward networks, can be trained using backpropagation, but, for the compressed-memory approaches, backpropagation provides only an approximation to the gradient. This is because the effective inputs to these networks include context patterns (previous hidden layer patterns) which depend in turn on inputs from earlier in the sequence. Backpropagation would have to proceed back to the beginning of the sequence to achieve a true gradient. In practice, however, the approximation which results from the cutoff seems to permit the effective learning of sequences as well as generalization to related novel sequences (Chater & Conkey, 1992).

The recognition network tested in the first set of experiments reported here is a simple recurrent network which takes phonetic segments as inputs and is trained to activate one unit each for the set of morphemes making up the input word. The network is shown in Figure 5. Note that the network also has an output layer trained to auto-associate the input. This has the effect of forcing the network to attempt to distinguish the different inputs, a prerequisite to using the short-term memory provided by the previous hidden layer (Servan-Schreiber, Cleeremans, & McClelland, 1988).

5.3 The Problem of "Semantics"

I have been assuming that there is a separate set of units for each category of morpheme to be recognized or produced. While in the version of the network shown in Figure 5 these output layers have identical connectivity with the network's hidden layer, the target for each layer is localized, so the network is provided with implicit information to the effect that there are separate output tasks, one for each morphological category. One category consists of content (lexical) morphemes, for example, verb roots such as EAT and SNORE or noun roots such as GIRL and TURNIP. Others are grammatical categories, for example, verb tense (PRESENT, PAST) or noun number (SINGULAR,

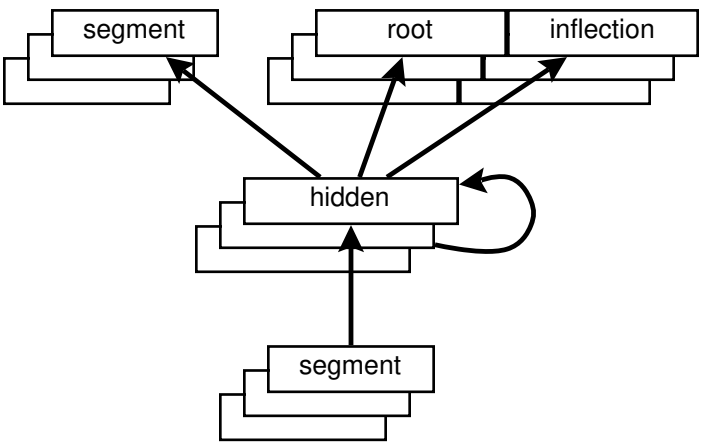

Figure 5: Recognition network

PLURAL). This is to be contrasted with a distributed approach to the representation of the meanings of words, which would fail to separate the lexical from the grammatical morphemes and the different grammatical categories from each other.

Thus the semantic component of the system which the morphophonological processor described here is a part of has somehow already distinguised lexical from grammatical "meaning" and localized the meanings that correspond to the roots of words. While this is not meant to imply that the brains of language learners come hard-wired for these semantic categories, how the system makes these distinctions is not explained in the model. The first distinction, that betweeen lexical and grammatical function, will take on more significance in the context of the modularized version of the model discussed below. The second, the localization of lexical meanings, has been adopted for the sake of simplicity only. It is not yet clear how the use of distributed representations for lexical (root) targets would affect the behavior of the network, beyond a general increase in difficulty.

5.4 Learning to Recognize Monomorphemic Words

A network similar to that shown in Figure 5, but with no inflection layer, was trained by Norris (1990) to recognize monomorphemic words. Norris shows not only that such a network is capable of performing the task but that it models some aspects of the time course of human word recognition. There are many questions that we might ask concerning the adequacy of the present approach for morphologically simple words, but I will not consider them in this paper. We shall return to the monomorphemic network later in the context of the learning of syllable representations for reduplication rules.

6 Learning to Recognize Polymorphemic Words

The major concern of this paper is how the proposed network performs on polymorphemic words, words formed through the application of productive rules which combine a set of morphemes.

6.1 Experimental Procedure

For all of the experiments reported in this paper, input words were generated randomly using an artificial language. The use of an artificial language permits one to control aspects of the phonology and morphology in a way that might be difficult or impossible with a real language. Unless otherwise indicated, each word in the language consisted either of a CVC or a CVCVC sequence ("C" = consonant, "V" = vowel), possible consonants were /p, b, f, v, m, t, d, s, z, n, k, g, x, ng/[9], and possible vowels were /i, e, a, u, o/. The final consonant of a word was limited to the set /p, f, m, t, s, k, x, ng/. Each phone was represented by a vector of 10 phonetic features: voice, vocalic, fricative, nasal, bilabial, dental, velar, high, back, and sonority. All features but sonority, which had five possible values, were binary.

Unless otherwise indicated, a set of 30 roots (15 CVC, 15 CVCVC) was generated randomly to use as stimuli for the recognition network. For each experiment, there was a set a set of one or more grammatical categories, each with two or more associated inflections. In most cases, there was a single category, which, to simplify matters, I will refer to as "tense", and two possible morphemes within that category ("present" and "past"). The number of possible word forms was thus 30 times the number of possible combinations of grammatical morphemes. In each case, the network was trained on 2/3 of the possible forms and tested on the remaining 1/3. This means, for example, for an experiment in which there are only two forms associated with each root, that is, two possible grammatical morphemes, that 1/3 of the roots (10) would be trained in both forms (20 words altogether), and the other 2/3 (20 roots) in only one of the two forms (20 words altogether). The test set would include the forms of these latter 20 which had not been trained. The set of roots to be tested and the forms in which they were to be tested were both selected randomly.

The output "meaning" target always consisted of a local pattern for each morpheme. For example, for an experiment in which there are two possible grammatical morphemes, each target consists of a vector of 32 values (30 roots and 2 grammatical morphemes), two of them 1.0, the rest 0.0.

In each experiment, the words were presented to the recognition network one segment at a time, in randomized order. Figure 6 illustrates the training procedure for the hypothetical sequence *ka*, the present tense of the verb *sing*. The input context layer (the previous hidden layer in the figure) was initialized to activations of .25 at the start of the word. Each word sequence was followed by an input pattern representing a word boundary ("#" in the figure), consisting of zero activations on all input units. The boundary was necessary to distinguish sequences such as *migu* and *miguk*; without the boundary the network would be unable at the end of the first word to know which of the two it was being presented. Following each forward pass, the hidden layer pattern was copied to the input context layer.

For any supervised classification problem, there are problems regarding what sort of target is appropriate. The most significant problem concerns the nature of targets for responses other than the correct one. This is related to the issue of negative evidence: while children may be told indirectly that a particular meaning is appropriate for a given form, they are not told at the same time that particular meanings are *not* appropriate for that form. However, with positive evidence alone, children are incapable of learning to recognize words. There are several possible solutions to this problem, most of them involving some sort of built-in mechanism which provides *implicit* negative evidence to the system. Among the simplest is simply to treat all responses other than the

[9] /ng/ represents a velar nasal.

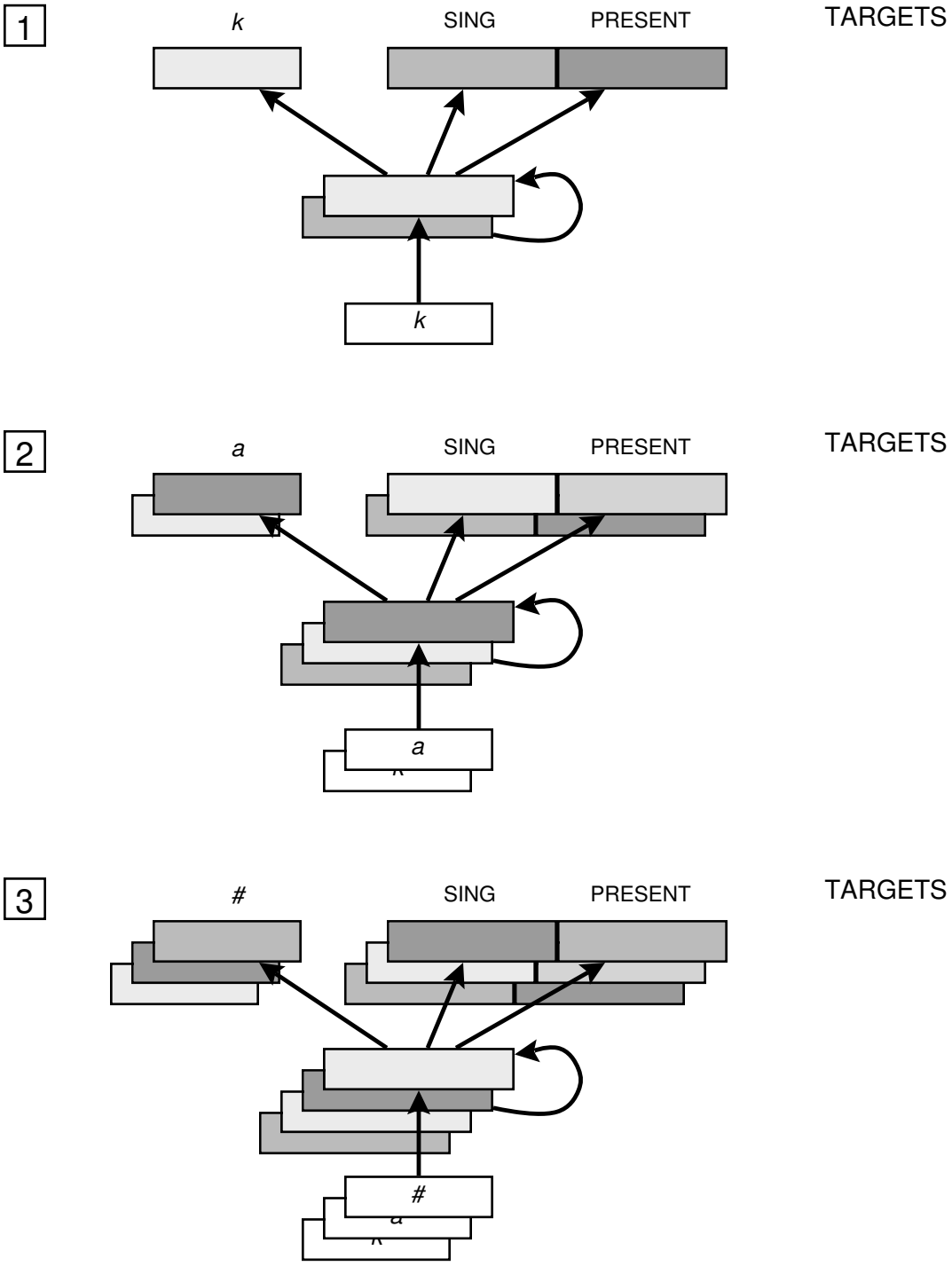

Figure 6: Training procedure for *ka*

correct one as wrong: for our localized "semantics", this means that the target for the correct unit is 1, that for all of the others 0. This implements a version of the **mutual exclusivity hypothesis**

(Markman, 1989), which has been applied to the learning of word meanings. Mutual exclusivity fails when a single input can map onto more than one output category, that is, homonymy in our case. Since this is relatively rare (and it will in any case not be considered in this paper), it may not present a serious problem.[10]

There is a related problem for *temporal* classification tasks, that of what target to provide during the presentation of the sequence. Early in the sequence, the sequence will often be ambiguous with respect to the output categories. At this point the correct target may tend to confuse the network because it weakens associations to all of the incorrect responses which are consistent with the sequence up to that point. On the other hand, anything other than a constant target seems implausible. In the experiments reported here, the root and inflection target remained constant throughout the presentation of the word.

The network was trained using conventional backpropagation. The error function was the hyperbolic arctan function suggested by Fahlman (1989), and Fahlman's (1989) "sigmoid prime offset" method was used to overcome the problem of the areas in the sigmoidal function where the derivative goes to zero. The learning rate was .1 and the momentum .5.

What interests us mainly is the extent to which the network generalizes about the rule or rules that it is trained on. This is measured by its performance on the set of test words. For each of these words, the network will have been trained on both the root form and each of the inflections but not on the particular combination that makes up the test word. For example, a test word might be the "past" of the root *migon* (formed according to whatever rule that network is being trained on). In this case, the network would have been trained on the "present" tense of this root and the "past" tense of others, but never on the past tense of *migon*.

Tests were conducted at regular intervals during training. Output "meaning" patterns were evaluated separately for the root and grammatical morpheme responses. Only the responses made at the end of each word, that is, following the final boundary marker, were evaluated because the network could not have been expected to respond correctly before the word has finished. A pattern was considered to be correct if it was closer to the correct root or grammatical morpheme pattern than to any other. Since there were 30 separate roots, the network had 1/30 of a chance of guessing the correct root on any given trial, and for the case where there were two possible inflections, the network had 1/2 a chance of guessing the inflection correctly.

Each experiment was run ten times, with different random weights.

6.2 Learning Morphological Rules in a Non-Modular Network

In the first set of experiments, the network shown in Figure 5 was trained on suffix, prefix, infix, circumfix, mutation, deletion, and template rules. In each case, each word consisted of two morphemes, a root and a single "tense" inflection, marking the "present" or "past". Examples of each rule:

- Suffix: present–*vibuni*, past–*vibuna*
- Prefix: present–*ivibun*, past–*avibun*
- Infix: present–*vikbun*, past–*vinbun*

[10] See Regier (1992) and Gasser & Smith (1993) for two approaches to negative evidence and mutual exclusivity when single categories for multiple inputs *are* frequent enough to present a problem.

- Circumfix: present–*ivibuni*, past–*avibuna*
- Mutation: present–*vibun*, past–*vibũn*
- Deletion: present–*vibun*, past–*vibu*
- Template: present–*vaban*, past–*vbaan*

For all but the template rules, the same set of 30 roots was used, each consisting of a CVC or CVCVC pattern. For the prefix, suffix, infix, and circumfix cases, both tense forms were marked by affixes consisting of a single segment (one prefix and one suffix segment in the case of circumfixes). This provides a stronger test of the network's ability to recognize the inflection than would be the case if only one form had the affix because the network cannot use the length of the word as a cue in this case. For both prefixes and suffixes, the two affixes were /i/ for the present and /a/ for the past. For circumfixes, the same affix preceded and followed the stem for each form. For infixes, a single consonant, /n/ for the present and /k/ for the past, followed the first vowel of the stem. For mutation, the present form was just the stem, and the past was formed by nasalizing the final vowel of the stem. For deletion, the present was again the stem, and the past was formed by deleting the final stem consonant. For the template rule, a set of 30 roots, each consisting of three consonants, was randomly generated. The present for each root was formed from the template $C_1aC_2aC_3$ and the past from the template $aC_1C_2aaC_3$.

Separate networks, each with 30 hidden units, were trained on each rule. Training proceeded for 150 epochs. Figures 7 and Figure 8 shows the results averaged over the ten trials for root and inflection recognition on the test words.

Performance on root recognition was much better than chance except in the case of circumfixation, and performance on inflection recognition was much better than chance in all cases.

Consider now the pattern of results for prefixing and suffixing. For the suffixing rule, performance was superior for the roots: the probability of achieving the network's performance by guessing was 10^{-21} for the inflections and 10^{-79} for the roots. The results for the prefixing rule are the reverse. Here it is the affix which is easier for the network: the tense of novel words is correctly identified in essentially all cases (probability of guessing at this rate: 10^{-49}) But recognition of the root of unfamiliar words is only three times the chance rate (probability of guessing at this rate: 10^{-4}).

What is it that leads to these asymmetries and the relatively poor performance, especially in the case of the roots in the prefixing case? Consider what aspects of the learning task would affect one or the other of the two subtasks, that is, the recognition of roots and inflections.

1. All else being equal, inflection recognition is easier than root recognition. This is because there are only two inflections to recognize, and each is only 1 (vs. 3 or 5) segment long.

2. A subsequence with a single invariant context during training and testing will have an advantage over one with a context which varies, all else being equal. This favors, for example, root recognition in the suffixing case over root recognition in the prefixing case because the context layer of the network is reinitialized at the start of each new word.

3. For a subsequence which appears during testing in the context of a novel previous subsequence, generalization should be encouraged by the appearance during training of a variety of contexts. This would favor inflection recognition in the suffixing case over root recognition

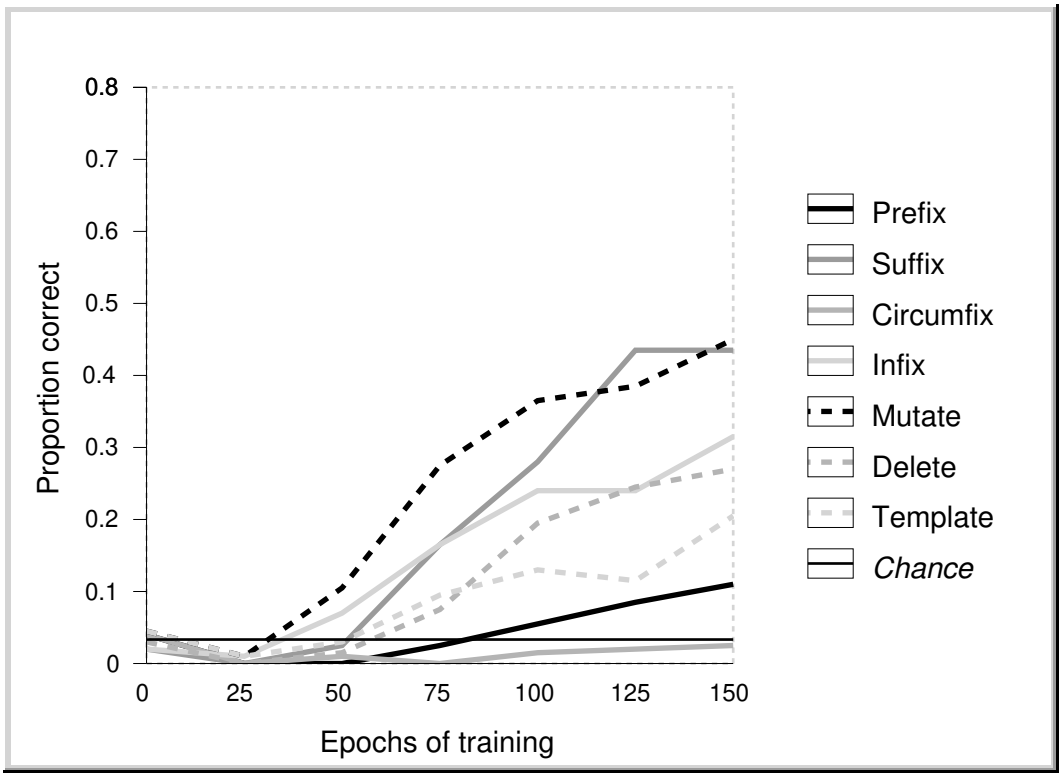

Figure 7: Root recognition by non-modular networks

in the prefixing case. Each of the two tense suffixes follows 20 different roots during training, so there is no strong association to either suffix from a particular subsequence. However, when the inflection is prefixed, each root which is tested appears during training following only one of the two prefixes. This should lead the network in effect to treat the prefix as part of the root. Then when the root is tested following the other prefix, the network may fail to identify it.

4. When a subsequence appears at or near the end of a word, the network receives at least some targets which are irrelevant to the identity of the subsequence. In the suffixing case, when a past-tense word is presented during training, the network is told that the form is past from the beginning of the word. Thus, for as many as five segments (the length of the stem), the network will adjust the weights into the two inflection output units on the basis of inputs which have nothing to do with the suffix which eventually appears. This should tend to favor prefixes over suffixes and word-initial over word-final stems.

5. When two tasks make conflicting demands on the network, the easier of the two tasks may in effect claim the hidden layer for itself, leaving performance on the harder task poorer than it would be otherwise. In both the prefixing and suffixing cases, the root recognition and inflection recognition tasks conflict with each other. In both cases the one task requires attention only to the beginnings of words while the other is performed best if the beginnings of words are ignored. However, in the prefixing case, the disparity in relative difficulty of

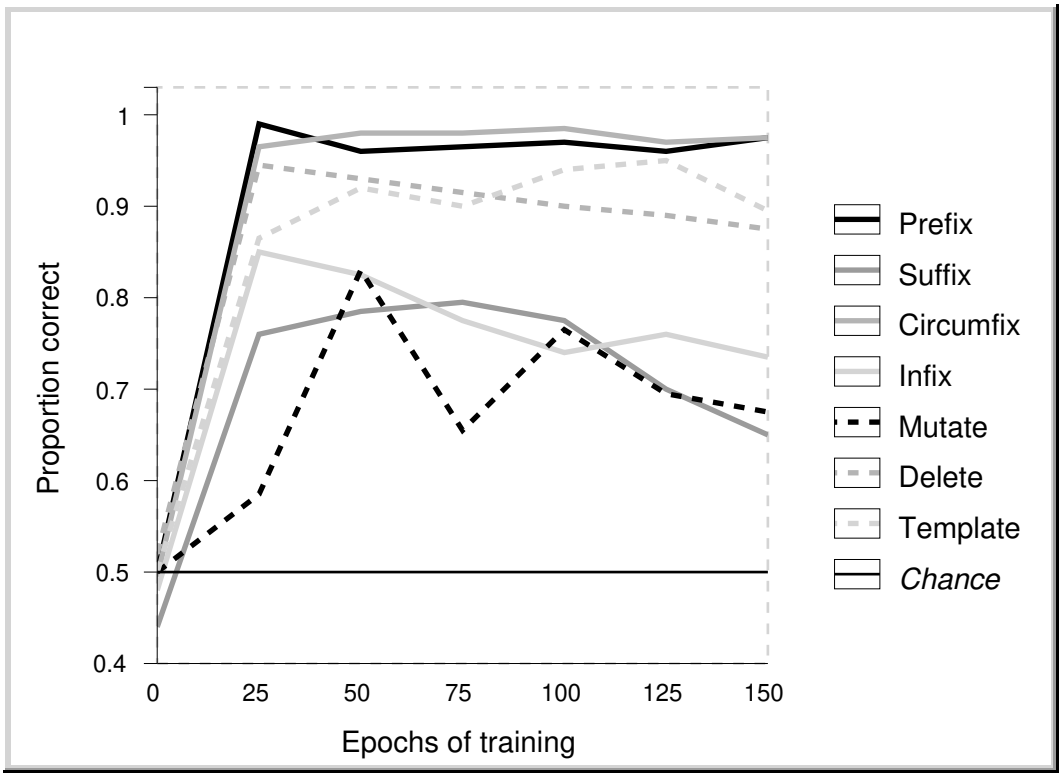

Figure 8: Inflection recognition by non-modular networks

the two tasks should be greater because of points 1, 2, and 4 above. That is, the prefixes are easy to recognize because they always appear in the same context, because they don't suffer from any irrelevant targets, and because there are fewer different forms to learn. Under these circumstances, the network should learn the prefixes very early in training, organizing itself in such a way that accurate root recognition is precluded.

On the basis of this *post hoc* account, the pattern of results for recognition by the network of prefixes, suffixes, and the stems they are affixed to is not surprising. But is there any way to alleviate any of these causes of poor performance? The last point, that of conflicting demands placed on the network, is a familiar one for backpropagation networks which are required to perform more than one task using a single hidden layer, the result being **crosstalk** (Jacobs, Jordan, & Barto, 1991). In **spatial crosstalk**, the network performs both conflicting tasks simultaneously; in **temporal crosstalk**, the tasks are performed at different times during processing. Our problem is an example of the former type since the network is expected on each time step to produce a response on both the root and grammatical morpheme output units.

To better visualize the problem, it helps to examine what happens in hidden-layer space as a word is processed. This 30-dimensional space is impossible to observe directly, but we can get an idea of the most significant movements through this space through the use of principal component analysis, a technique which is by now a familiar way of analyzing the behavior of recurrent networks (Elman, 1991; Port, 1990). Given a set of data vectors, principal component analysis yields a set of

orthogonal vectors, or components, which are ranked in terms of how much of the variance in the data they account for.

Principal components for the hidden layer vectors were extracted for a single recognition network before training and following 150 repetitions of the prefix training patterns. Figures 9 and 10 show the paths traced by the hidden layer patters along components 1 and 2 (the two components which account for the most variance in hidden layer patterns) as the words *ipomum*, *apomum*, *ingesos*, and *angesos* are processed by the network before and after training. Note that these words include pairs with the same roots and pairs with the same inflections. Before training the paths for the words with common roots are nearly indistinguishable; the single segment distinguishing the two sequences is not enough to make much difference in the response along these components. Following training, however, it is differences in inflection that dominate the paths. In particular, component 2 is clearly dedicated to the prefix recognition task. While there is some movement through this space as the root appears, words with common roots end up further apart than those with different roots.

In the next section I describe a modified architecture which addresses the problem of the conflict between root and inflection recognition.

6.3 Learning Morphological Rules in a Modular Network

6.3.1 A Modular Architecture for Recognition

One obvious answer to the problem of two conflicting tasks in a single network is to assign the tasks to separate networks, or separate modules within one network (Jacobs et al., 1991). A network with separate hidden layers for each task would obviate crosstalk, resulting ultimately in better generalization for both of the tasks. In the suffixing case, for example, if one portion of the network is devoted to the problem of recognizing the root only, it can make heavy use of context, as is required for root recognition. If there is another set of hidden units responsible for recognizing tense, this portion of the network can learn to make less use of context. These units can also concentrate on detecting the particular segment(s) representing the suffix and ignore others, while the root hidden layer units must be aware of all possible input phones which can appear in roots. In the prefixing case, the hidden-layer units responsible for the relatively simple task of recognizing prefixes, a task that requires attention to only the first two segments of a word, can be prevented from interfering with the units faced with the more difficult task of learning to recognize roots, which involves attending to the last three or five segments of a word.

Figure 11 shows a modular architecture for the recognition of words consisting of a root and a single grammatical morpheme. Each weight responds to error on one or the other of the two recognition tasks, but not to both as in the non-modular case. Thus we are really dealing here with two completely separate networks, though I will continue to refer to the entire system as a "network".

A modular network with the architecture shown in Figure 11 was trained on the same rules used in the experiments with the non-modular networks. To permit comparison with the non-modular network, the modular network also had 30 hidden units, divided evenly between the two modular hidden layers. Note that there are thus fewer overall connections in the modular case. Results for test trials are shown in Figures 12 and 13.

Note first that generalization improves for all rule types for both inflection and root recognition. Next I consider each of the rule types in turn. We will be concerned in each case with isolating the aspects of the task which affect the difficulty of root or inflection recognition.

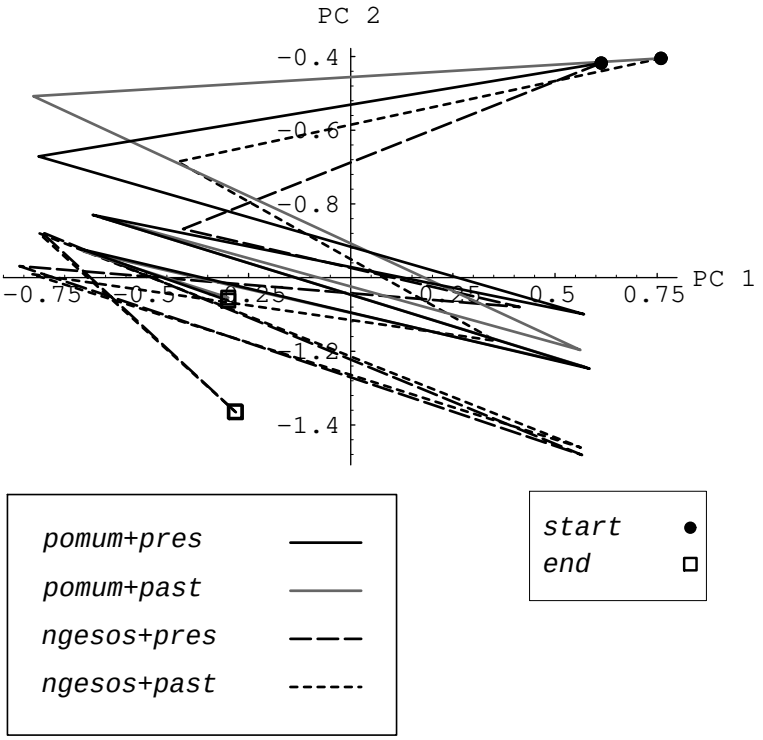

Figure 9: Prefix rule, untrained network, Principal Components 1 and 2, *i-pomum, a-pomum, i-ngesos, a-ngesos*

6.3.2 Prefixation and Suffixation

Consider the results for suffixation and prefixation. Performance improves in particular for suffix recognition, for which performance is close to perfect.

For prefixation there is also dramatic improvement in generalization over the non-modular architecture for the more difficult task, that of recognizing the root. The probability of guessing right at the rate of the network is now 10^{-53}.

Again principal component analysis can clarify what is going in the networks. Separate analyses were conducted for the modular hidden layers of a network trained on the prefixing rule. Figures 14

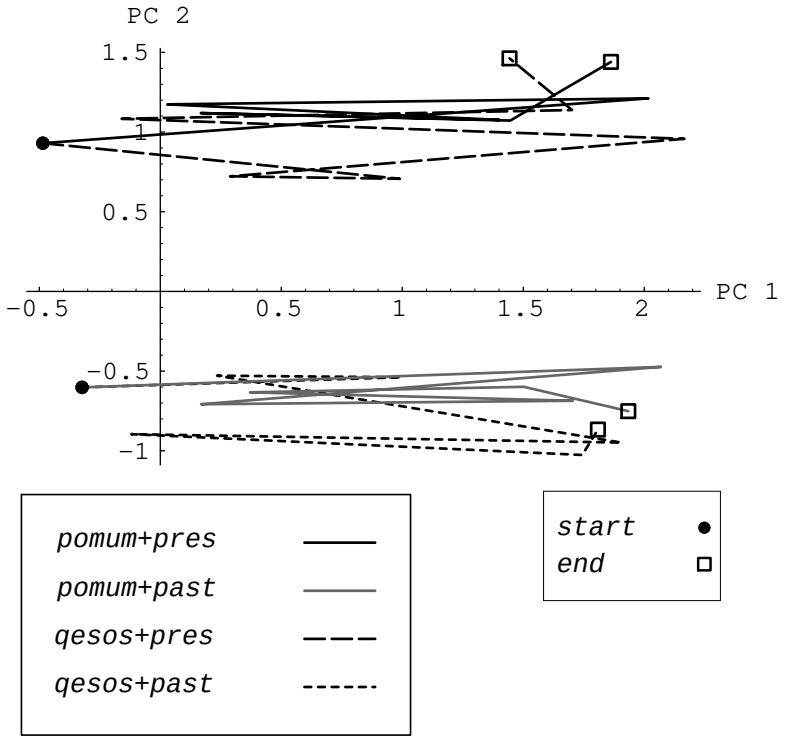

Figure 10: Prefix rule, trained network, Principal Components 1 and 2, *i-pomum, a-pomum, i-ngesos, a-ngesos*

and 15 show the paths traced by the network along the first and second principal components within the two modules as it processed the same four words shown in Figure 9 and 10. In the inflection module, words with the same inflection end up in the same region in the space defined by these two components (it is component 1 which predominates for inflection recognition), while in the root module, words with the same root end up in the same region.

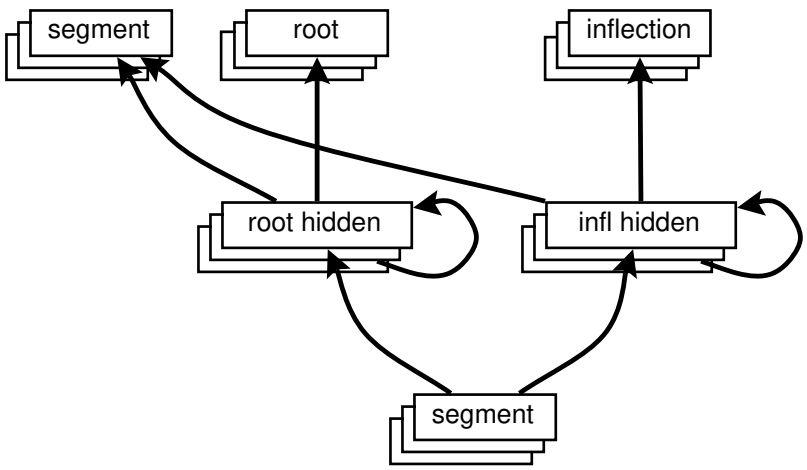

Figure 11: Modular architecture for recognition

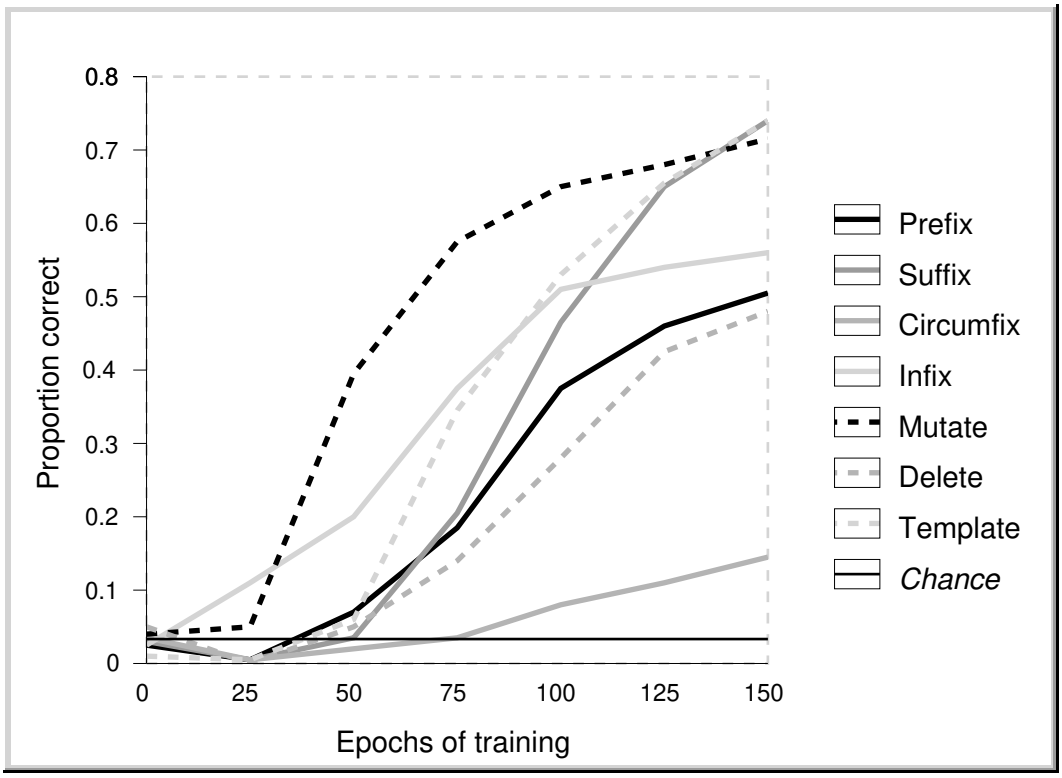

Figure 12: Root recognition by modular networks

6.3.3 Circumfixation

Circumfixation is really just prefixation and suffixation combined. With respect to root recognition, this means that performance should be no better than in the prefixation experiment because, as with prefixation, a test root appears in a single previous context during training, but a different one during

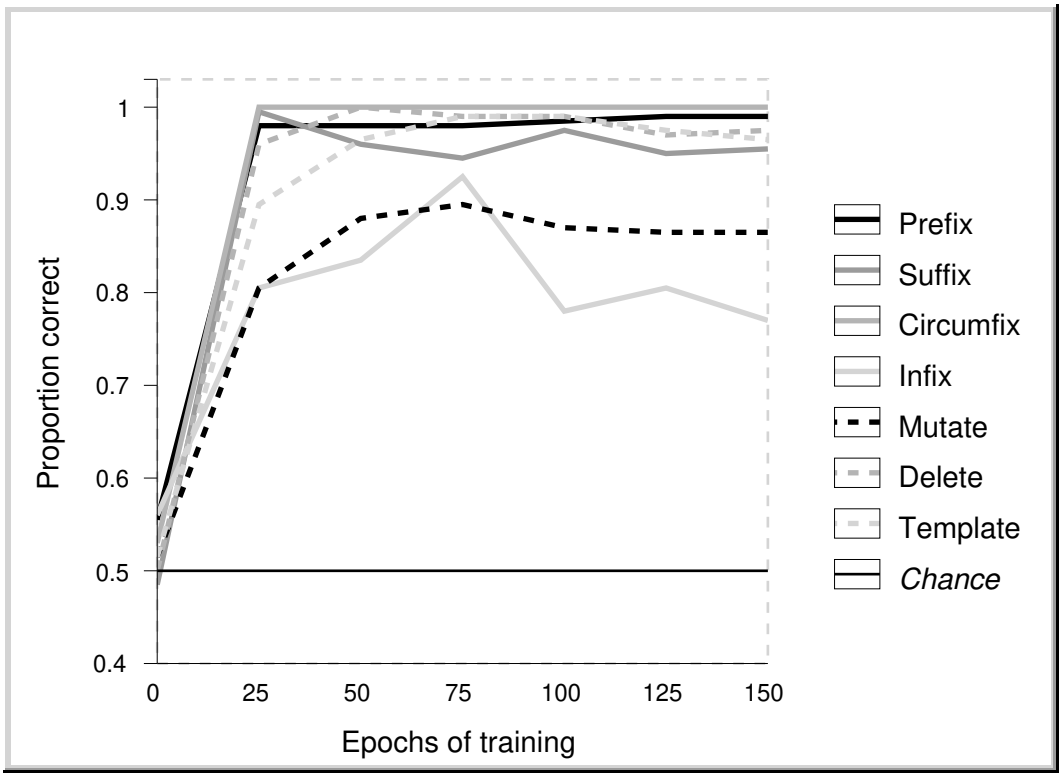

Figure 13: Inflection recognition by modular networks

testing. Performance on inflection recognition, on the other hand, should be higher than in either the prefixation or suffixation case because here there are redundant cues to the tense of the input word. The predictions are confirmed regarding both root and inflection recognition. The extremely poor performance on root recognition is apparently due to the conflicting demands placed on the root module; the network needs to ignore both the beginnings and the ends of words.

6.3.4 Infixation

For infixation, the inflection appears within the stem rather than on either end of it. Languages tend to be consistent in where infixes appear within the stem. As noted above, these positions may be defined with respect to the beginnings or the ends of the stems.

To investigate the effects of infix position on recognition, three experiments in addition to the infixation experiment already reported were conducted. In one the infix appeared in a position which was a constant distance from the beginning of each stem ("pre-infix"), in another the infix appeared in a position which was a constant distance from the end of each stem ("post-infix"), and in the third the position of the infix varied from one stem to another ("mixed infix"). In each case, words consisted of either CVC, CVCVC, or CVCVCVC syllables, and there were three different inflections ("present", "past", and "future"), consisting of the single-segment infixes /n/, /k/, and /f/. In the pre-infix case, the infix was the second consonant of the word (CV<u>C</u>, CV<u>C</u>VC, CV<u>C</u>VCVC); in the post-infix case, the infix was the second from the last consonant (<u>C</u>VC, CV<u>C</u>VC, CV<u>C</u>VC),

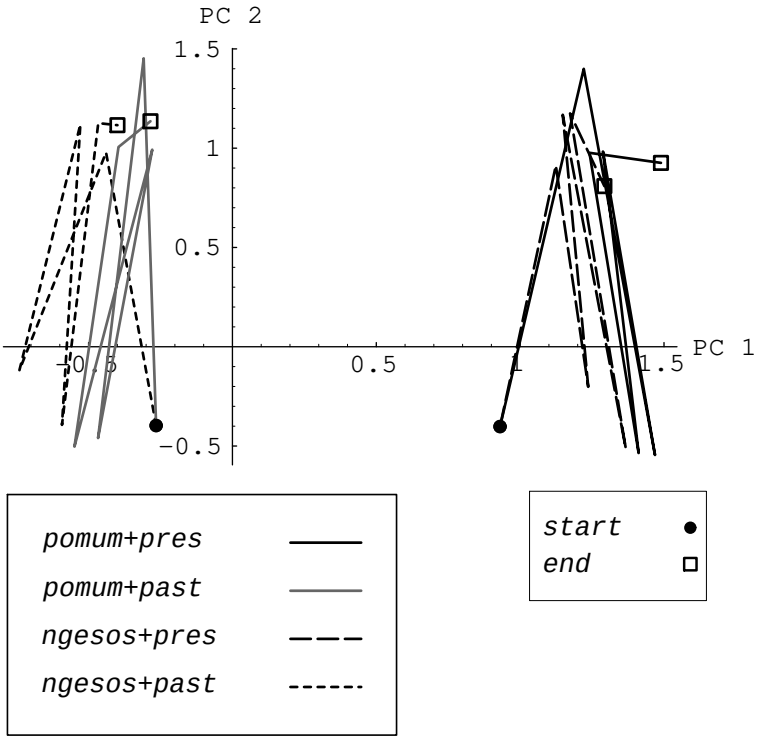

Figure 14: Prefix rule, inflection module, Principal Components 1 and 2, *i-pomum, a-pomum, i-ngesos, a-ngesos*

and in the mixed case, half of the roots had pre-infixes, the other half post-infixes. To simplify root recognition somewhat, the vowels in each root were constrained to be the same. Results are shown in Figure 16.

Consider first root recognition. Overall, the results indicate that root recognition is relatively easy for this task. This is probably due to the fact that there are three separate forms ("tenses") for each root. There are two possible ways that a child or a network might learn to recognize roots which include infixes. One would involve learning at what position in the word the infix appears and then attending only to the other positions in the word when identifying the root. In other words, the child or network would in a sense be learning a sort of template for the root. This

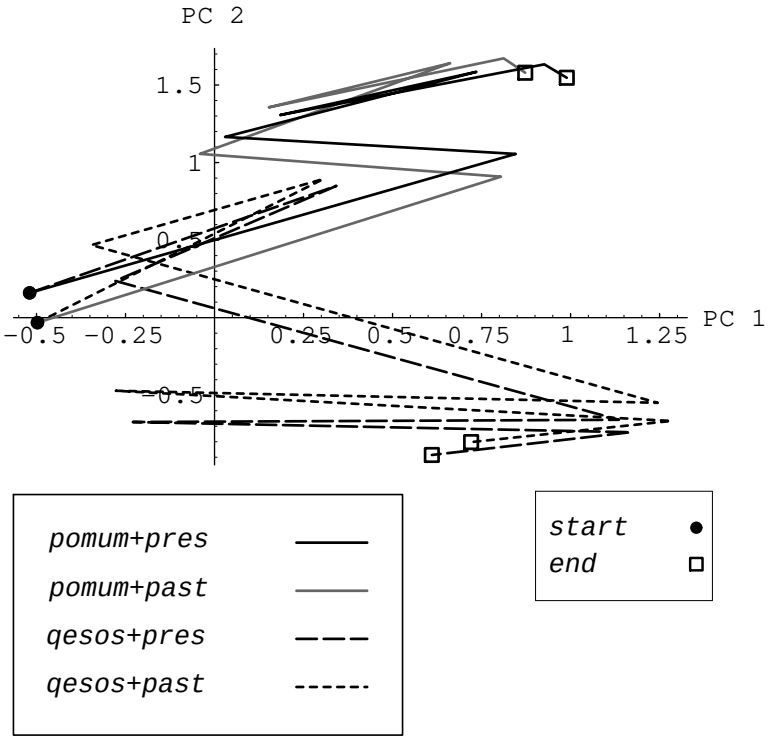

Figure 15: Prefix rule, root module, Principal Components 1 and 2, *i-pomum, a-pomum, i-ngesos, a-ngesos*

strategy would of course require that the infix always appear in the same position, as it normally does in natural languages. The other strategy would be simply to associate consistent sequences of segments, wherever they appear in the word, with particular roots, in a sense treating the task as though there were no infixes to worry about. If the network were using only this second strategy, then performance on the "mixed" task would not be significantly worse than that on the "pre-infix" and "post-infix" case because there would be no benefit to having the infixes in consistent positions. At least in comparing the "post-infix" to the "mixed" case, this is in fact what we find. The network does not learn to associate particular positions defined in terms of their distance from the end of the word with the portion that is to be ignored in identifying the root.

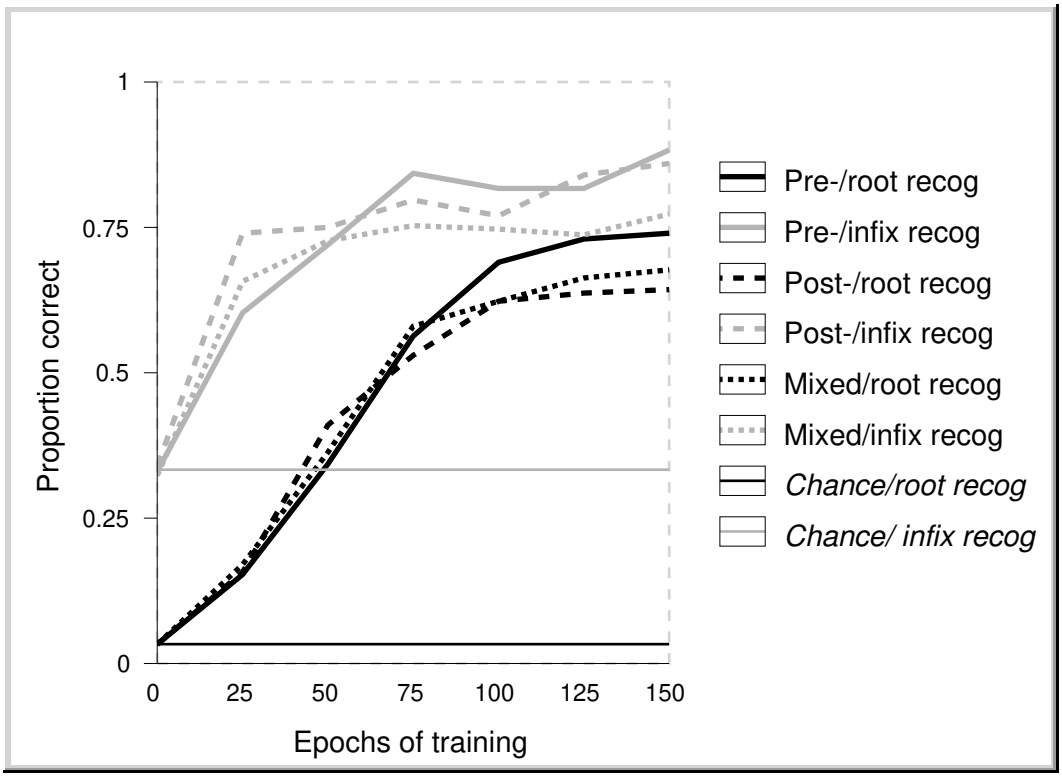

Figure 16: Pre-, post-, and mixed infixation rules, modular networks

Note, however, that the results are better in the "pre-infix" case. This could mean that the network is making use of the first strategy above, that is, that it has learned to "count" from the beginning of the words to the consistent position where the infix occurs. Or it could mean simply that the network benefits from the fact that more often when a word finishes, there is a subsequence of at least length two in the context memory which belongs entirely to the root. This would mean using the second of the strategies mentioned above. An examination of the network's errors shows that errors are most frequent for the CVC words, that is, those where the infix is actually a suffix. This is one indication that the network is profiting from the second of the two strategies rather than the first. That is, where the word ends in a subsequence which belongs to the root, that is, in the CVCVC and CVCVCVC cases, there are fewer errors. Other evidence for this conclusion is provided by principal component analysis, which shows that for the same root in different tense forms, the network overcomes the effect of the irrelevant (infix) segment by bringing the positions of the words in hidden-layer space closer and closer as the segments are presented. Figure 17 shows the path traced by the root recognition module of the network through the space defined by the first and second principal components as the three different forms of a single root, *bo_onot*, are presented to the network. Of course the paths are the same for the first two segments of the words. When the third infix segment, that is, either /n/, /k/, or /f/, is reached, the paths diverge considerably. Then, through the remaining portions of each word, the paths again approach each other so that, by the end of the word, they are in the same region of the space, enabling the same response at the output. Thus, rather than ignoring the infix segment, the network benefits from the fact that in this case four

consecutive segments are enough to distinguish the root from all others.

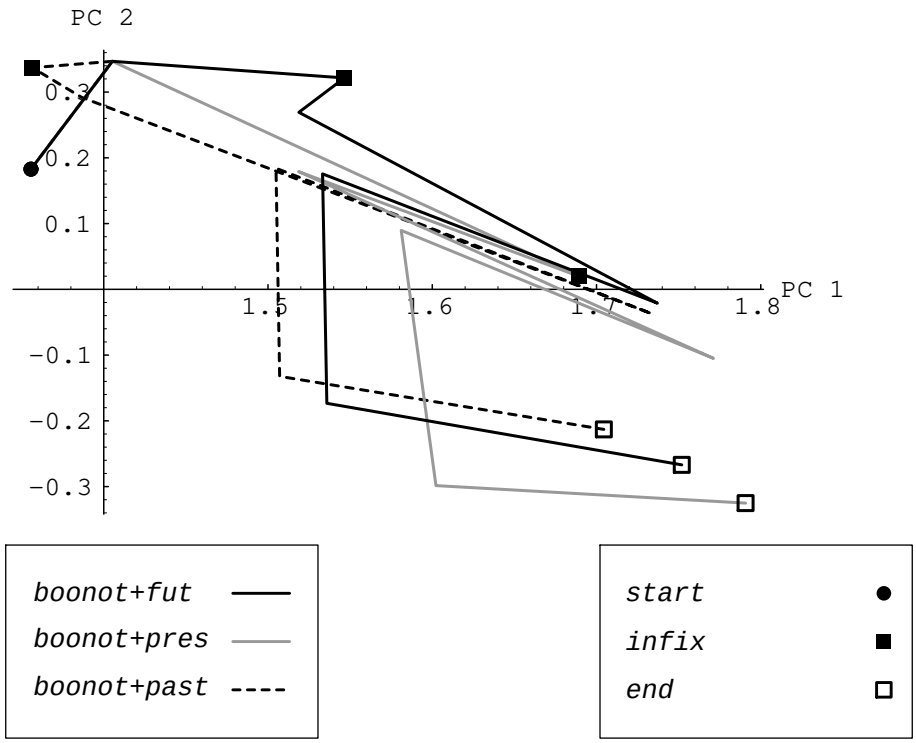

Figure 17: Infixation rule, root module, Principal Components 1 and 2, *bononot, bokonot, bofonot*

In the case of recognition of the infix itself, the second strategy above would seem to be ruled out. Since the infix segment is always one which could occur elsewhere in the word, it is not enough to simply look for that segment; the network must know where to look. In order to solve the problem, the inflection recognition module must learn to count. Thus it is not surprising that performance degrades considerably in the "mixed" case when the position of the infix varies.

6.3.5 Mutation

Mutation is similar to infixation in that the portion of the stem which is mutated may be anywhere within it, and performance is similar for the two types of rules: relatively high for root recognition and low for inflection recognition.

To better understand how the network solves mutation tasks, three additional experiments, similar to the infixation experiments, were run for mutation. In all cases, the words consisted of CVC and CVCVC syllables, and the mutation consisted in the nasalization of one of the vowels in the word. In the "pre-mutation" experiment, it is the first vowel, in the "post-mutation" the last vowel, and in the "mixed" case, one or the other of the two vowels which was mutated.

Results for these experiments are shown in Figure 18. For mutation, the difference in the two forms that the network sees for each root can be seen as an *addition* to the same form, so there is no need, as with infixation, to ignore the segment which is the locus of the mutation. Thus it is not surprising that for root recognition, the results for the "mixed" case are not significantly lower than those for the two cases where the mutation is in a constant position within the word. Similarly, for recognition of the inflection, it is not necessary to pay attention to a particular position within the word because nasalization occurs only in the "past" tense form. Thus if a word contains a nasalized segment anywhere, it is "past"; otherwise, it is "present". Still it is considerably easier for the network when the relevant segment is earlier rather than later in the word. Why this is so is not clear.

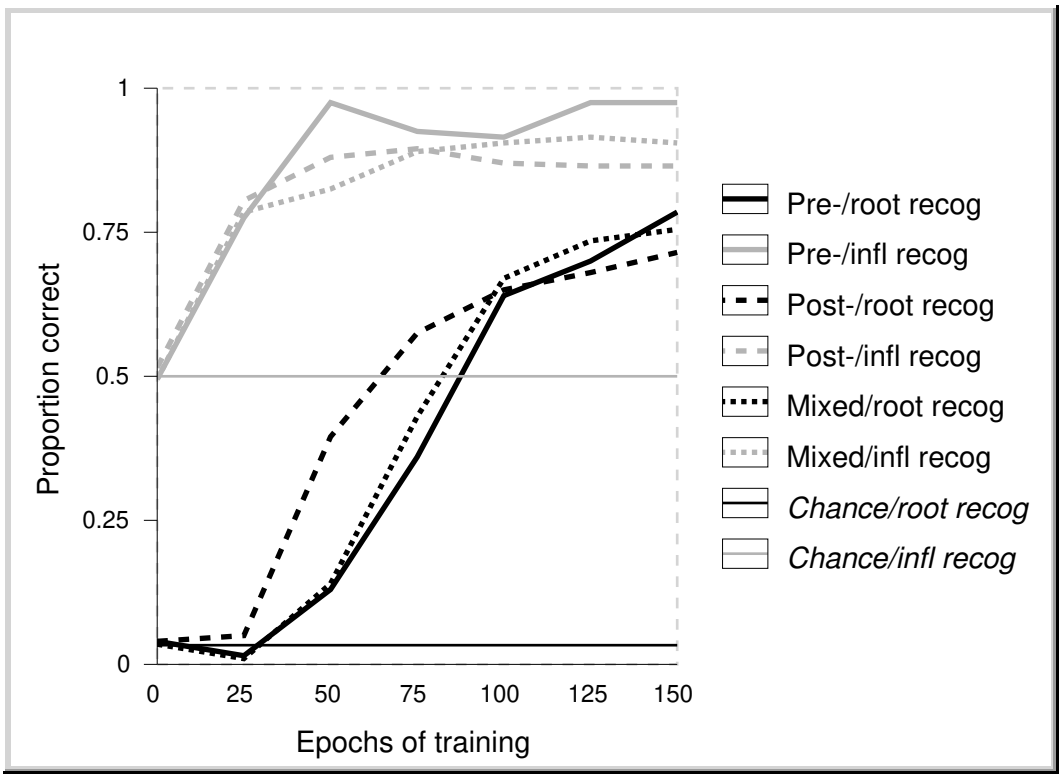

Figure 18: Pre-, post-, and mixed mutation rules, modular networks

Rules normally thought of as mutation rules in natural languages are often not of the type used here: mutation may result in a segment or sequence of segments which can also occur elsewhere in words not subject to the mutation rule. This is true, for example, of the past tense of strong verbs in Germanic languages (*drink*/*drank*). But rules of this sort could also be viewed as infixation, as discussed above. Thus the English verb *drink* would be seen as consisting of the root *drnk* which takes the infixes *i* and *a* for the present and past forms.

6.3.6 Deletion

The deletion rule tested in these experiments is somewhat similar to suffixation because the difference in the two forms is located at the end of the word, so it useful to compare performance here with that on suffixation. Root recognition is poorer for deletion than for suffixation. This is understandable because there is an inherent difficulty with deletion rules. The problem comes in recognizing a word which has been previously encountered in the deleted form (here the past tense) only. Under these circumstances, there is no way of knowing what the deleted segment is. When tested on the corresponding form without deletion, the network is expected to respond with the appropriate root meaning following a final consonant which it has not been trained to associate with the root. (Recall that it is the output of the recognition at the end of the word which is evaluated.) It should not be surprising that deletion rules are extremely rare in natural languages. Note, however, that inflection recognition is better for deletion than for suffixation. This is probably due to the nature of the particular forms used in the experiment. All present forms end in a consonant and all past forms in a vowel, so it is enough for the inflection module to determine what sort of segment a form ends in to identify its tense.

6.3.7 Templatic Rules

For the templatic rule experiment, the two forms of each root shared the same initial and final consonant. This tended to make root recognition relatively easy; it is among the highest of the rules examined. With respect to inflections, the pattern is more like infixation than prefixation or suffixation because all of the segments relevant to the tense, that is, the /a/s are between the first and last segment. Inflection recognition is also very high, probably because of the redundancy: the present tense is characterized by an /a/ in second position and a consonant in third position, the past tense by a consonant in second position and an /a/ in third position.

To gain a better understanding of the way in which the network solves a template morphology task, a further experiment was conducted. In this experiment, each root consisted of a sequence of three consonants from the set {*p*, *b*, *m*, *t*, *d*, *s*, *n*, *k*, *g*}. There were three tense morphemes, each characterized by a particular template. The present template was $C_1aC_2aC_3a$, the past template $aC_1C_2aaC_3$, and the future template $aC_1aC_2C_3a$. Thus the three forms for the root *pmn* were *pamana*, *apmaan*, and *apamna*. The network learns to recognize the tense templates very quickly; generalization is over 90% following only 25 epochs of training. This task is relatively easy since the vowels appear in the same sequential positions for each tense. More interesting is the performance of the root recognition module, which must learn to recognize the commonality among sequences of the same consonants even though, for any pair of forms for a given root, only one of the three consonants appears in the same position. Performance is 72% on the test words following 150 epochs.

It is possible to get an idea of how the network solves this task by again examining the principal components of hidden-layer space. The paths through the space defined by the first two components of the root recognition module as the three forms of the root *pds* are presented to the network are shown in Figure 19. Points marked in the same way represent the same root consonant.[11] What we see is that, as the root-recognition module processes the word, it passes through roughly similar regions in hidden-layer space as it encounters the consonants of the root, independent of their sequential position.

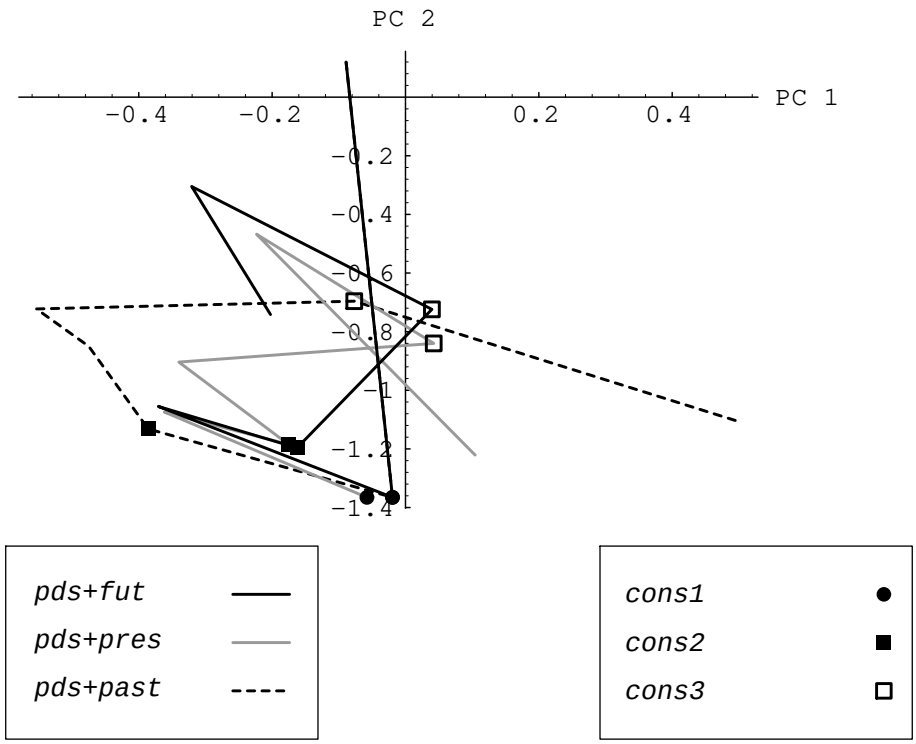

Figure 19: Templatic rule, root module, Principal Components 1 and 2, *padasa, apdaas, apadsa*

[11] Only two points appear for the first root consonant because the first two segments of the past and future forms of a given root are the same.

6.4 Evaluating Modularity

We can view the modular network investigated in the last set of experiments in two ways, either as a analytical device for teasing apart the different properties of root and inflection recognition or as a proposal for a model of the learning of word recognition in children. In this section, I consider some of the implications of the latter alternative.

The modular approach to the recognition of polymorphemic words is clearly superior to the non-modular alternative, and the reasons for this are clear: the root and inflection recognition tasks are in conflict, so assigning them to different portions of the network results in better performance on both tasks. But there are a number of questions that the modular approach raises. First, it is important to be clear on the nature of the modularity being proposed here. As discussed above, I have defined the task of word recognition in such a way that there is a built-in distinction between lexical and grammatical "meanings" because these are localized in separate output layers. The modular architecture of Figure 11 extends this distinction into the domain of morphophonology. That is, the shape of words is represented internally (on the hidden layer) in terms of two distinct patterns, one for the root and one for the tense, and the network "knows" this even before it is trained.

A second question concerns precisely what the modules are to be responsible for. This becomes an issue when we consider what happens when more than one grammatical category is represented in the words being recognized. As noted above, it is not uncommon in many languages for words to be composed of five or more morphemes. With respect to modularity, there are two options for such cases.

1. There is a separate hidden layer module for each grammatical category, as well as for roots. A network for recognizing Swahili verbs, for example, would require at least 5 separate modules, one for the verb stem and one each for the subject, tense, object, and mood markers. The number of modules thus depends on the language being learned.

2. There is a fixed number of modules which are shared among the output tasks presented by the target language.

The first alternative requires either a large set of modules which can be recruited as they prove necessary for the language being learned or a mechanism for creating modules from a fixed set of hidden-layer units as they become necessary. In this extreme form, this option would preclude any sharing at the hidden layer among the various tasks, preventing any phonological generalizations across different morphological categories. A more reasonable variant of the first alternative would provide for separate modules for each output morphological category but at the same time leave one set of hidden units with connections to all output groups. The multi-purpose layer could be responsible for generalizations that cut across the different categories, for example, generalizations about the syllable structure of the target language. While this is an appealing possibility, I will not consider it further in this paper.

The second alternative, a fixed set of modules to be shared among the output tasks, is somewhat simpler to implement. But how many modules should there be, and how are they to be shared? Ideally, the network would have the capacity to figure out for itself how to distribute the modules it starts with among the various output tasks presented by the target language; I return to this possibility below. But it is also informative to investigate what sort of a sharing arrangement achieves the best performance. For example, given two modules and three output tasks, root recognition and

the recognition of two separate inflections, which of the three possible ways of sharing the modules achieves the best performance? We might expect the highest performance with an arrangement involving sharing between the two inflection recognition tasks because this would build into the network something like the distinction between lexicon and grammar which is fundamental to many linguistic and psycholinguistic models.

Two sets of experiments were conducted to investigate the optimal use of fixed modules by a recognition network, one designed to determine the best way of distributing modules among output tasks when the number of modules does not match the number of output tasks and one designed to determine whether a network could assign the modules to the tasks itself. In both sets of experiments, the stimuli were words composed of a stem and two affixes, either two suffixes or one prefix and one suffix. Thus there were two morphological categories, say "tense" and "aspect", each represented by two different morphemes. The roots were the same ones used in the affixation and deletion experiments already reported. In the prefix–suffix case, the two prefixes were /u/ and /e/ and the two suffixes /a/ and /i/. Thus the four forms for the root *migon* were *umigona*, *umigoni*, *emigona*, and *emigoni*. In the two-suffix case, the first suffix was /a/ or /i/, the second suffix /s/ or /k/. Thus the four forms for the root *migon* were *migonik*, *migonis*, *migonuk*, and *migonus*. There were in all cases two hidden-layer modules.

Since there were two modules and three output tasks (one root and two inflections), there were three different ways to divide the tasks among the modules. In each case one module was shared by two tasks, while the other module was dedicated to one task.

6.4.1 Which Sort of Modularity?

In the first set of experiments, these three possibilities were compared for each of the two rule types (prefix and suffix, two suffixes). A pilot experiment with a separate module for each of the three output tasks determined that good performance was achieved with a hidden layer of 20 units for root recognition and hidden layers of 3 units each for the two inflection recognition tasks. Therefore, the hidden layer modules used in these experiments were of the following sizes: (1) shared module: connected to root and first affix output layers, 23 units; task-specific module: connected to second affix output layer, 3 units; (2) shared module: connected to root and second affix output layers, 23 units; task-specific module: connected to first affix output layer, 3 units; (3) shared module: connected to both affix output layers, 6 units; task-specific module: connected to root output layer, 20 units. Each network was trained for 100 epochs and tested every 10 epochs.

The results for the two types of affixation are shown in Figures 20 and 21. Lines are labeled according to the tasks which are shared by one of the modules and by the task for the results are shown. For root recognition there is a clear advantage in both cases to the arrangement in which neither affix recognition task shares hidden units with the root recognition task. This is not surprising because, as we have already seen, both prefix and suffix recognition interfere with root recognition, and since these tasks are learned faster, they tend to "take over" the units in the layer responsible for them. What these experiments make clear is that, even though the affix recognition tasks are easily learned with only 3 units, when they are provided with more units (23 in these experiments), they will tend to "distribute" themselves over the available units. If this were not the case, performance on the competing, and more difficult, task, root recognition, would be no better when it has 20 units to itself than when it shares 23 units with one of the other two tasks.

The results for affix recognition for the two affixation types are shown in Figures 20 and 21.

The results are clear in the two-suffix case: the arrangement in which the affix recognition tasks share a module is again superior. In other words, the network does somewhat better at recognizing either of the two suffixes when a single layer of 6 units is responsible for both tasks than when a separate layer of 3 units is responsible for each. The former arrangement is a more efficient one because of what the two suffix recognition tasks share. In the prefix–suffix case, the results are not so clear. The situation in which a single module is shared by both affix recognition tasks is somewhat superior to the other alternatives, but not by as great a margin as in the two-suffix case. Apparently, from the perspective of the network, prefixing and suffixing do not have as much in common as do two suffixes. It is difficult to conclude from these results whether the modularity that is called for for the word recognition task corresponds to the conventional lexicon-grammar division.

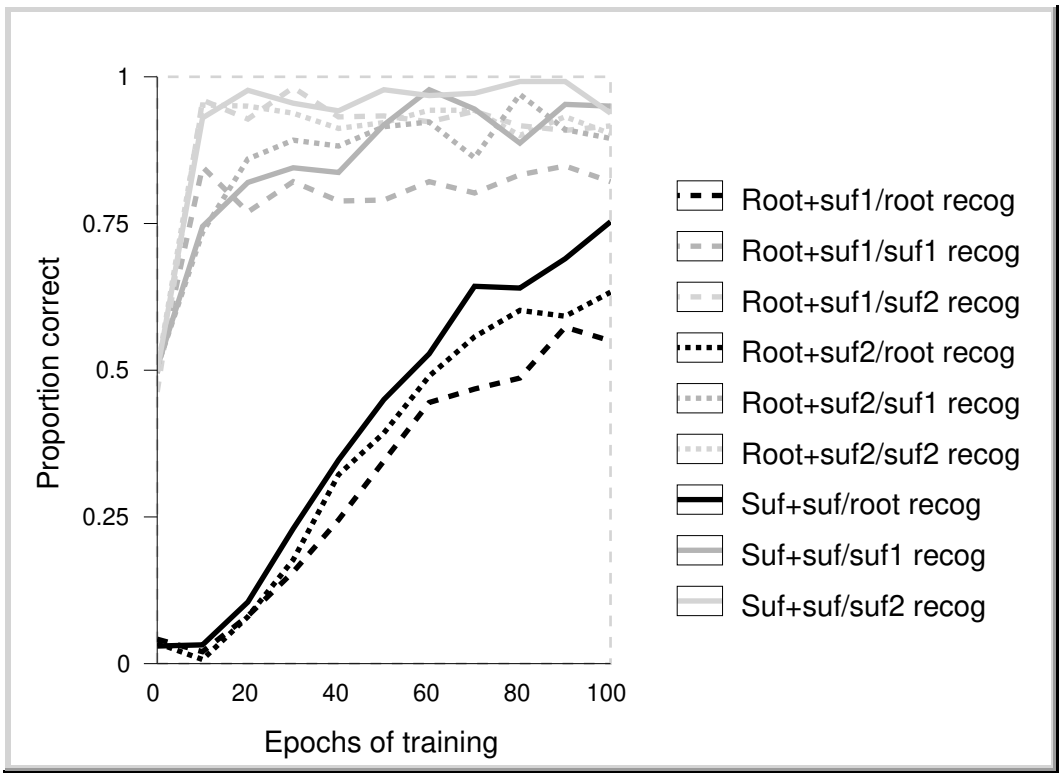

Figure 20: Two-suffix rules, modular networks

6.4.2 Adaptive Use of Modules

If one distribution of the available modules is more efficient than the others, we would like the network to be able to find this distribution on its own. Otherwise it would have to be wired into the system from the start. What this would amount to would depend on what the modularity we are concerned with actually gains the system, and the results from the last section are not sufficient to tell us this. If, for example, the most efficient arrangement is one along the lines of the lexicon-grammar distinction, that is, one which treats all of the inflections within a single module, the system would

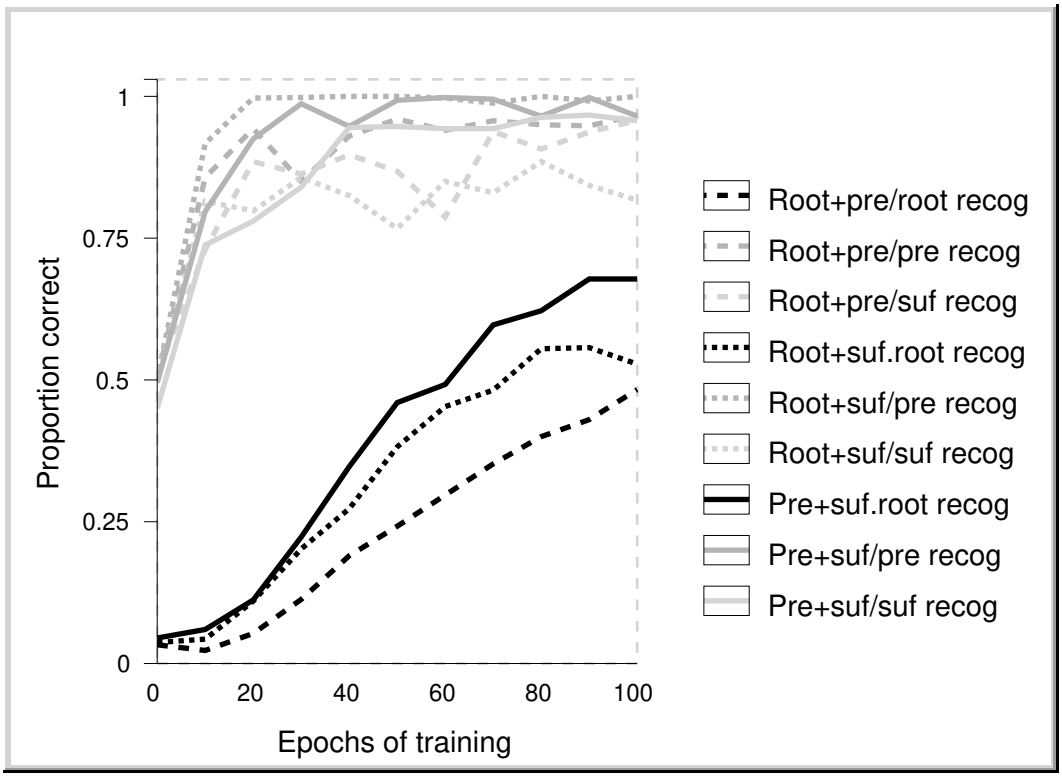

Figure 21: Prefix and suffix rules, modular networks

need to somehow know, for example, that tense belongs in the same general category as subject person/number for a language such as Swahili where these are marked on the verb with separate inflections. Confusing this process is the fact that some notions such as SIZE may be signalled by grammatical morphology as well as lexical items in one language but by lexical items alone in another. If, on the other hand, efficient use of modules has something to do with where an inflection appears in a word, the network would need to know before training that certain inflections will take the form of prefixes and others the form of suffixes, an obviously implausible state of affairs. In either case, some form of *adaptive* use of the available models is clearly called for.

Given a system with a fixed set of modules but no wired-in constraints on how they are used to solve the various output tasks, can a network organize itself in such a way that it uses the modules efficiently? There has been considerable interest in the last few years in architectures which are endowed with modularity and learn to use the modularity to solve tasks which call for it. The architecture described by Jacobs et al. (1991) is an example. In this approach there are connections from each modular hidden layer to all of the output units. In addition there are one or more gating networks whose function is to modulate the input to the output units from the hidden-layer modules. In the simplest version of the architecture, which is appropriate for domains in which temporal crosstalk is a problem, there is a single gating network output unit for each module. The outputs of the modules are weighted by the outputs of the corresponding gating units to give the output of the entire system. The whole network is trained using backpropagation. For each of the modules, the error term is what it would be in a non-modular system, except that the error is weighted by the

value of the gating input as it is passed back to the modules. Thus each module adjusts its weights in such a way that the difference between the system's output and the desired target is minimized, and the extent to which a module's weights are changed depends on its contribution to the output, that is, the amount of input from the corresponding gating network. For the gating networks, the error function is more complicated. On a given trial, if one module sufficiently outperforms the others, the error is minimized when the output of the gating unit for the winning module approaches 1.0, the outputs of the other gating units approach 0.0, the total output of the gating units approaches 1.0, and the gating outputs are binary. If no module is a clear winner, error is minimized when all gating outputs go to a neutral value. The effect of the error function for the gating networks is to implement competition among the modules for each output task group.

For the version of the architecture that is appropriate for problems involving spatial, as opposed to temporal, crosstalk, as in the present case, there is a single gating unit responsible for the set of connections from each hidden module to each output task group. The error function is the same except that the determination of whether there is a winning module proceeds separately for each of the output groups. For our purposes, two further augmentations are required. First, we are dealing with recurrent networks, so we permit each of the modular hidden layers to see its own previous values in addition to the current input, but not the previous values of the hidden layers of the other modules. Second, we are interested not only in competition among the modules for the output groups, but also in competition among the output groups for the modules. In particular, we would like to prevent the network from assigning a single module to all output tasks. To achieve this, the error function is modified so that error is minimized, all else being equal, when the total of the outputs of all gating units dedicated to a single module is neither close to 0.0 nor close to the total number of output groups.

Figure 22 shows the architecture for the situation in which there is only one inflection to be learned. The connections ending in circles symbolize the competition between sets of gating units which is built into the error function for the network. Note that the gating units have no input connections. These units have only to learn a bias, which, once the system is stable, leads to a relatively constant output. The assumption is that, since we are dealing with a spatial crosstalk problem, the way in which particular modules are assigned to particular tasks should not vary with the input to the network.

The first experiment with the adaptive modular network was designed to insure that for the single grammatical morpheme case, the network would in fact assign separate modules to the two output tasks, that is, the root and grammatical morpheme. Networks of the type shown in Figure 22 with two modules of 15 units each were trained on the prefixing rule used in the experiments described above. Following 120 epochs of training, the outputs of the two gating units associated with each model were averaged over a single epoch. These four average weights provide a measure of how the network has divided the modules between the output tasks. In ten separate runs, each network organized itself in such a way that one module was assigned to the root and one to the prefix; that is, for one module, the gating unit with the higher average output was associated with the prefix, and for the other module, the gating unit with the higher average output was associated with the root.

Next a set of experiments tested whether the adaptive modular architecture would assign two modules to three tasks (root and two inflection) in the most efficient way for the two-suffix and prefix-suffix cases. Recall that the most efficient pattern of connectivity in both cases was the one in which one of the two modules was shared by the two suffix recognition tasks.

Adaptive modular networks with two modules of 15 units each were trained on the two-suffix

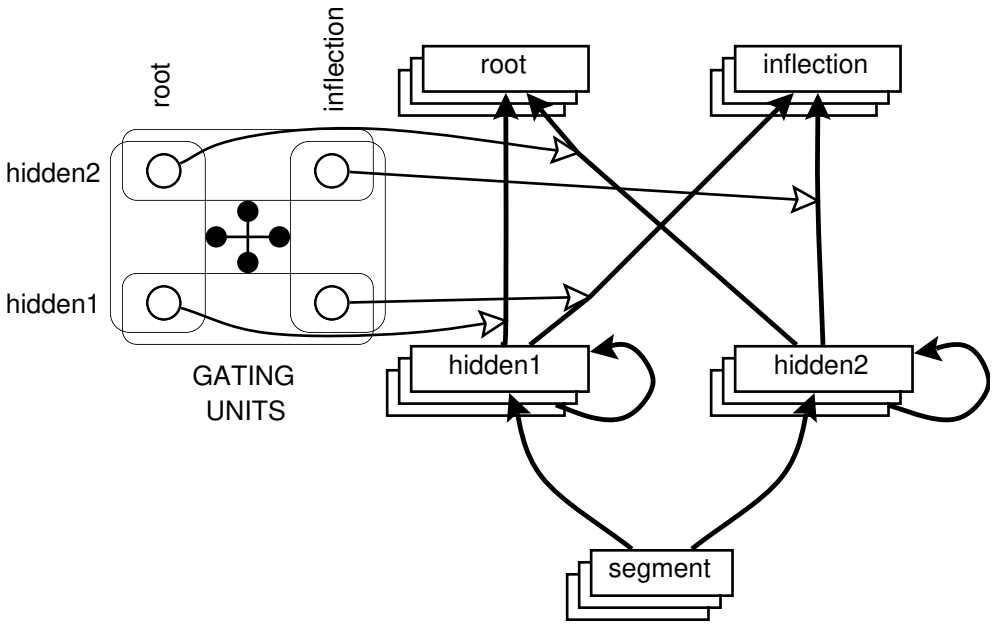

Figure 22: Adaptive modular architecture for recognition

and prefix-suffix tasks described in the last section. Again, following 120 epochs, the average outputs of the six gating units for the different modules were examined to determine how the modules were shared. The results were negative; the three possible ways of assigning the modules to the three recognition tasks occurred with approximately equal frequency. The problem was that the inflection recognition tasks were so much easier than the root recognition task that they claimed the two modules for themselves early on, while neither module was strongly preferred by the root task. Thus as often as not, the two grammatical morphemes ended up assigned to different modules.

This suggests ways to give root recognition an advantage over inflection recognition. It is well-known that children begin to acquire lexical morphemes before they acquire grammatical morphemes. Among the reasons for this is probably the more abstract, less salient nature of the meanings of the grammatical morphemes. In terms of the tasks faced by the network, this relative difficulty would translate into an inability to recognize what the grammatical morpheme targets would be for particular input patterns. Thus we could model this by delaying training on the grammatical morphemes.

The experiment with the adaptive modular networks was repeated, this time with the following training regimen. Entire words (consisting of root and two affixes) were presented throughout training, but for the first 80 epochs, the network saw targets for only the root recognition task. That is, the connections into the output units for the two inflections were not altered during this phase. Following the 80th epoch, by which time the network was well on its way to recognizing the roots, training on the inflections was introduced. Following the This procedure was followed for both the two-suffix and prefix-suffix tasks; 20 separate networks were trained for each type. For the two-suffix task, in all cases the network organized itself in the predicted way. That is, for all 20 networks one of the modules was associated mainly with the two inflection output units and the other associated with the root output units. In the prefix-suffix case, however, the results were more

equivocal. Only 12 out of 20 of the networks organized themselves in such a way that the two inflection tasks were shared by one module, while in the 8 other cases, one module was shared by the root and prefix recognition tasks.

The difference is not surprising when we consider the nature of the advantage of the configuration in which the two inflection recognition tasks are shared by one module for the two categories of words. In both cases, roots are recognized better with this configuration. But this will have little effect on the way the network it organizes itself because, following the 80th epoch when competition among the three output tasks is introduced, one or the other of the modules will already be firmly linked to the root recognition layer. At this point, the outcome will depend mainly on the competition between the two inflection recognition tasks for the two modules, the one already claimed for root recognition and the one which is still unused. Thus we can expect this training regimen to settle on the best configuration only when it makes a significant difference for inflection, as opposed to root, recognition. Since this difference was greater for the two-suffix words than for the prefix-suffix words, there is a greater preference in the two-suffix case for the configuration in which the two inflection tasks are shared by a single module. It is also of interest that for the prefix-suffix case, the network never chose to share one module between the root and the suffix; this is easily the least efficient of the three configurations from the perspective of inflection recognition.

We would expect different results, of course, if the grammatical morphemes were trained before the root, but this would go against the facts of language acquisition.

Thus we are left with only a partial solution to the problem of how the modular architecture might arise in the first place. For circumstances in which the different sorts of modularity impinge on performance on inflection recognition, the adaptive approach can find the right configuration. When it is performance on root recognition that makes the difference, however, this approach has nothing to offer. Future work will have to address what happens when there are more than two modules and/or more than two grammatical morphemes in a word.

6.5 Reduplication

We have yet to deal with reduplication, which presents a special challenge because of the need to recognize not only that something has been added to the stem but that it is a copy of some portion of the stem. In what follows, I will touch upon only a few of the many issues that reduplication brings up, and the present model only offers the barest beginnings of an account of the recognition of words with reduplication.

Consider the recognition of a novel form in which some portion has been reduplicated and for which the system is familiar with the corresponding form in which there is no reduplication. Clearly this process involves recognizing the similarity between the relevant portions of the unfamiliar word. Thus rather than actually attempt to teach a network a reduplication rule, I will be concerned with the somewhat simpler task of determining whether a network can recognize similarities between succeeding stretches of segments.

Reduplication operates frequently at the syllable level, and it is only syllable reduplication that I will be dealing with here. For the simple recurrent networks we have considered so far, recognition of reduplication would seem to be a difficult, if not an impossible, task. Consider the case in which a network has just heard the sequence *tuta*. At this point we would expect a human listener to be aware that the two syllables had the same first consonant. The process seems to require a direct comparison between representations for two syllables. But at the point following the *a*, the network

does not have access to representations for the two subsequences.

It is possible to train a simple recurrent network on the simplest of reduplication rules. For example, presented with CVCV sequences, and trained to turn on an output unit whenever the first and second consonants are identical, a simple recurrent network generalizes easily. As demonstrated by Corina (1991), it is also possible, though difficult, to train a simple recurrent network to *produce* words embodying a reduplication rules considerably more complex than this.

To test the capacity of a simple recurrent network of the type we have been applying to word recognition, networks were trained consisting of two syllables each, where the initial consonant ("onset") of each syllable came from the set /p, b, f, v, m, t, d, s, z, n, k, g, x, gh, ng/[12], the vowel from the set /i, e, u, o, a/, and the final consonant, when there was one, from the set /n, s/. Separate networks were trained to turn on their single output unit when the onsets of the two syllables were the same and when the "rimes", that is, everything but the onset, were the same.

The training set consisted of 200 words. In each case, half of the sequences satisfied the reduplication criterion. Results of the two experiments are shown in Figure 23 by the lines marked "Seq". Clearly these networks failed to learn this relatively simple reduplication task. While these experiments do not prove conclusively that a recurrent network, presented with words one segment at a time, is incapable of learning reduplication, it is obvious that this task is not an easy one for these networks.

In a sequential network, input sequences are realized as movements through state space. It appears, however, that recognition of reduplication requires the explicit comparison of *static* representations of the subsequences in question, e.g., for syllables in the case of syllable reduplication. If a simple recurrent network like the ones we have seen thus far is trained to recognize, that is, to distinguish, the syllables in a language, then the pattern appearing on the hidden layer following the presentation of a syllable must encode all of the segments in the syllable. It is, in effect, a summary of the sequence that is the syllable.

It is a straightforward matter to train a network to distinguish all possible syllables in a language. We simply treat the syllables as separate words in the monomorphemic word recognition network, that is, one like that shown in Figure 11 but without the inflection output layer.

A network of this type was trained to recognize all 165 possible syllables in the same artificial language used in the experiment with the sequential network. When presented to the network, each syllable sequence was followed by a boundary segment consisting of zeroes.

The hidden-layer pattern appearing at the end of each syllable-plus-boundary sequence was then treated as a static representation of the syllable sequence for a second task. Pairs of these syllable representations, the same one used to train the sequential network in the previous experiment, were used as inputs to two simple feedforward networks, one trained to respond if its two input syllables had the same initial consonant, the other trained to respond if the two inputs had the same rime (that is, whatever follows same vowel, that is, the same rules trained in the previous experiment.[13] Again the training set consisted of 200 pairs of syllables, the test set of 50 pairs in each case. Results of these experiments are shown in Figure 23 by the lines labeled "FF". Although performance is far from perfect, it is clear that these networks have made the appropriate generalization. This means that the syllable representations encode the structure of the syllables in a form which enables the relevant comparisons to be made.

What I have said so far about reduplication, however, falls far short of an adequate account.

[12] /gh/ represents a voiced velar fricative.

[13] For these networks, the learning rate was .05.

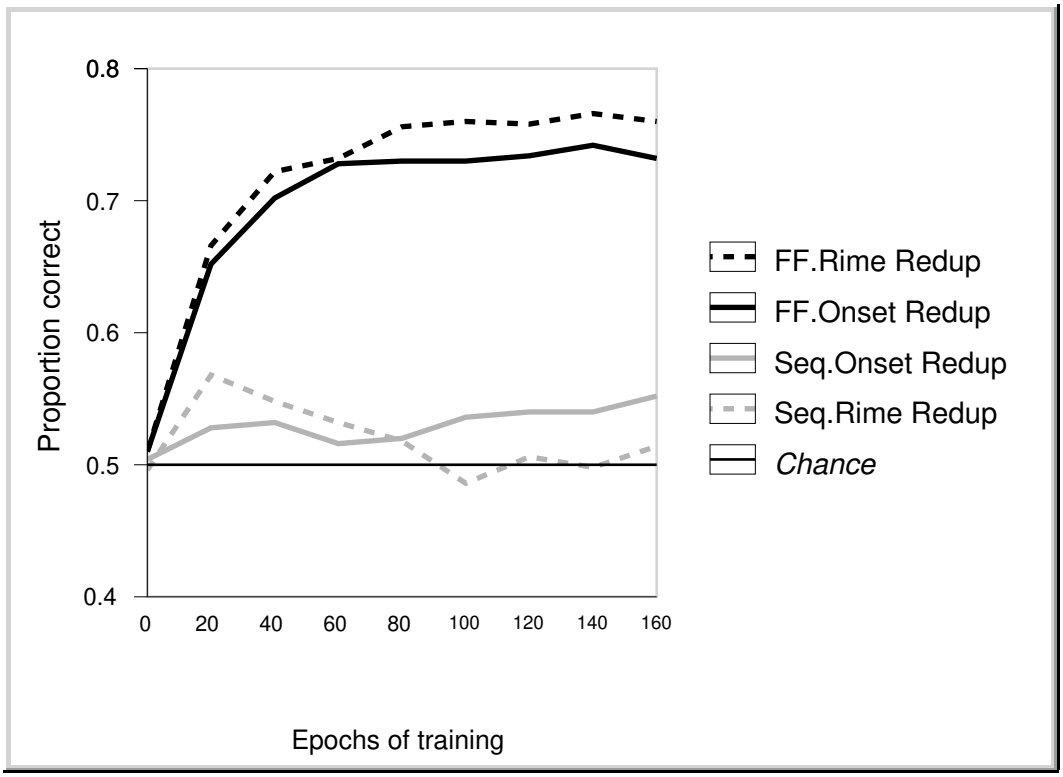

Figure 23: Reduplication rules, sequential and feed-forward networks trained with distributed syllables

First, there is the problem of how the network is to make use of static syllable representations in recognizing reduplication. That is, how is access to be maintained to the representation for the syllable which occurred two or more time steps back? Second, for this approach to work at all, the network, or some mechanism outside the network, must be capable of breaking words into their component syllables as they enter the recognition system.

For syllable representations to be compared directly, a portion of the network needs to run, in a sense, in syllable time. That is, rather than individual segments, the inputs to the relevant portion of the network need to be entire syllable representations. Combining this with the segment-level inputs that we have made use of in previous experiments gives a hierarchical architecture like that shown in Figure 24. In this network, word recognition, which takes place at the output level, can take as its input both segment and syllable sequences. The segment portion of the network, appearing on the left in the figure, is identical to what we have seen thus far. (Hidden-layer modularity is omitted from the figure to simplify it.) The syllable portion, on the right, runs on a different "clock" from the segment portion. In the segment portion activation is passed forward and error backward each time a new segment is presented to the network. In the syllable portion this happens each time a new syllable appears. Just as the segment subnetwork begins with context-free segment representations, the syllable subnetwork takes as inputs context-free syllables. This is achieved by replacing the context (that is, the recurrent input to the SYLLABLE layer) by a boundary pattern at the beginning

of each new syllable.[14]

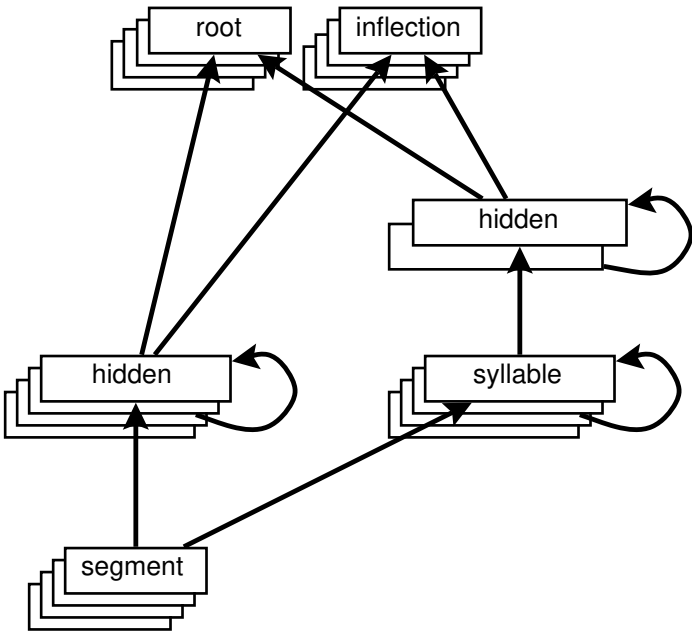

Figure 24: Hierarchical recognition architecture

But there remains the question of how the network is to know when one syllable ends and another begins. Again this is a topic to which entire academic careers might be devoted, so the answer suggested here will be a simplistic one. One of the features of input segments provided to the networks in the experiments reported here is **sonority**. Sonority, for those who believe in it (e.g., Selkirk, (1982)), is roughly a reflection of the extent to which the vocal apparatus is open during the production of a segment. Syllable structure, it has been argued, can be described in terms of the rises and falls of sonority. The **peaks** of syllables, typically vowels, are the segments with the highest sonority, and the segments preceding the peak, the syllable **onset**, show increasing sonority as they approach the peak, while those following the peak if there are any, the syllable **coda**, decrease in sonority. For purposes of segmentation, the major problem is that a consonant following a vowel may belong either to the syllable containing the vowel or to the following syllable. In most languages, the former is true if another consonant follows the consonant, the latter if a vowel follows. The problem is that it is therefore in general impossible, when receiving segments one at a time from left to right, to know whether the current segment belongs to the current syllable or to the next one. This does not mean that it is impossible to parse segments into syllables; listeners seem to accomplish this. But this makes syllabification a difficult task for a sequential network with no way to backtrack. The tentative solution offered here is the standard one in cases where ambiguous situations call for deterministic solutions: lookahead. If the decision on where to syllabify is postponed until sonority rises beyond a certain threshold, it is possible to syllabify appropriately under most circumstances. Figure 25 shows an architecture that would accomplish this. The figure shows a portion of the network in Figure 24 together with a layer designed to detect

[14]A more appealing possibility is a contextual input which includes traces of previous context; this would implement *relative* context-freeness of phonological representations.

syllable boundaries. The connections from the segment to the syllable layer have a delay of one time step (one segment, that is). Thus the syllable boundary layer is seeing the segment that is ahead of what is currently being input to the syllable layer. The syllable boundary layer responds when both sonority is rising and it crosses a certain threshold. At this point it causes the syllable context (input from its pattern on the previous time step) to be interrupted so that the new syllable begins with an empty context, and at the same time it causes the representation of the previous syllable, that is, the pattern on the syllable layer, to be sent on to higher layers. In this way, it controls the "clock" of the syllable portion of the network. For the purposes of this paper, I assume that this syllabification network is already in place. In reality much of it would need to be learned or at least tuned, perhaps during an early pre-semantic phase of development. How this might happen is beyond the scope of this paper though.

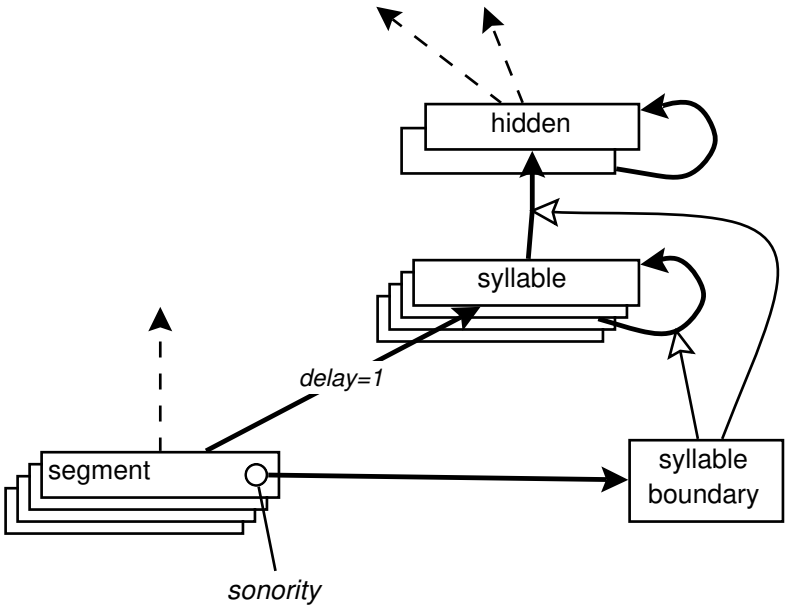

Figure 25: Syllabification network

As noted above, reduplication may operate at levels other than the syllable. For example, it may be defined in terms of total or partial copying of entire multisyllabic morphemes. If we accept the arguments made in this section, the implication is that what I have said about syllables should also apply to higher-level units of phonological or morphological organization. Thus there would be a process of division into units and a portion of the network which treats one of these units at a time as a primitive input. One such level which seems plausible is that of metrical feet (Hogg & McCully, 1987), multisyllabic units defined in terms of the patterns of stress on the syllables. Division into such units might be possible on the basis of stress, much as division into syllables may be based on sonority.

Syllables and higher-level units are motivated for reduplication. But there are other reasons for believing that they are psychologically real. Syllables seem to be necessary, for example, to support the process of stress assignment in production.

More important for our own purposes is yet another motivation for some sort of higher-level

unit. The representations of syllables or other subsequences which appear on the hidden layer of the basic network are candidates for the intermediate representations which provide the link between perception and production. In fact, in a series of experiments to be described in another paper and referred to briefly in Gasser (1992), the syllable representations learned by a recognition network have been shown to support the learning of production for all of the types of rules investigated in this paper.

6.6 Constraints on Morphological Processes

In the previous sections, I have described how modular simple recurrent networks have the capacity to learn to recognize morphologically complex words resulting from a variety of morphological processes. But is this approach too powerful? Can these networks learn rules of types that people cannot? While it is not completely clear what rules people can and cannot learn, some evidence in this direction comes from examining large numbers of languages. One possible constraint on morphological rules was mentioned above, the constraint which, in the terms of autosegmental analyses, states that association lines do not cross.

Can a recognition network learn a rule which violates this constraint as readily as a comparable one which does not? To test this, separate networks were trained to learn the following two template morphology rules, involving three forms, which I will refer to as "present", "past", and "future".

1. present: $C_1aC_2aC_3a$, past: $aC_1C_2aaC_3$, future: $aC_1aC_2C_3a$
2. present: $C_1aC_2C_3aa$, past: $aC_1C_2aC_3a$, future: $aC_1aC_3aC_2$

Both rules produce the three forms of each root using the three root consonants and sequences of three *a*'s. In each case each of the three consonants appears in the same position in two of the three forms. The second rule differs from the first in that the order of the three consonants is not constant; the second and third consonant of the present and past forms reverse their relative positions in the future form. In the terms of a linguistic analysis, the root consonants would appear in one order in the underlying representation of the root (preserved in the present and past forms) but in the reverse order in the future form. The underlying order is preserved in all three forms for the first rule. I will refer to the first rule as the "favored" one, the second as the "disfavored" one.

In the experiments testing the ease with which these two rules were learned, a set of thirty roots was again generated randomly. In this case each root consisted of three consonants limited to the set: {p, b, m, t, d, n, k, g}. The networks used had modularity like that shown in Figure 11. As before, the networks were trained on 2/3 of the possible combinations of root and grammatical morpheme (60 words in all) and tested on the remaining third (30 words). Results are shown in Figure 26. While the results do not show a dramatic difference, there is a clear advantage for the favored over the disfavored rule with respect to generalization for root recognition. Since the grammatical morpheme ("tense") is easily recognized by the pattern of consonants and vowels, the order of the second and third root consonants is irrelevant to grammatical morpheme recognition. Root recognition, on the other hand, depends crucially on the sequence of consonants. With the first rule, in fact, it is possible to completely ignore the CV templates and pay attention only to the root consonants in identifying the root. With the second rule, however, the only way to be sure which root is intended is to keep track of which sequences occur with which templates. With the two possible roots *ftn* and *fnt*, for example, there would be no way of knowing which root appeared in a form not encountered during training unless the combination of sequence and tense had somehow

been attended to during training. In this case, the future of one root has the same sequence of consonants as the present and past of the other. Thus, to the extent that roots overlap with one another, root recognition with the disfavored rule presents a harder task to a network. Given the relatively small set of consonants in these experiments, there is considerable overlap among the roots, and this is reflected in the poor generalization for the disfavored rule.

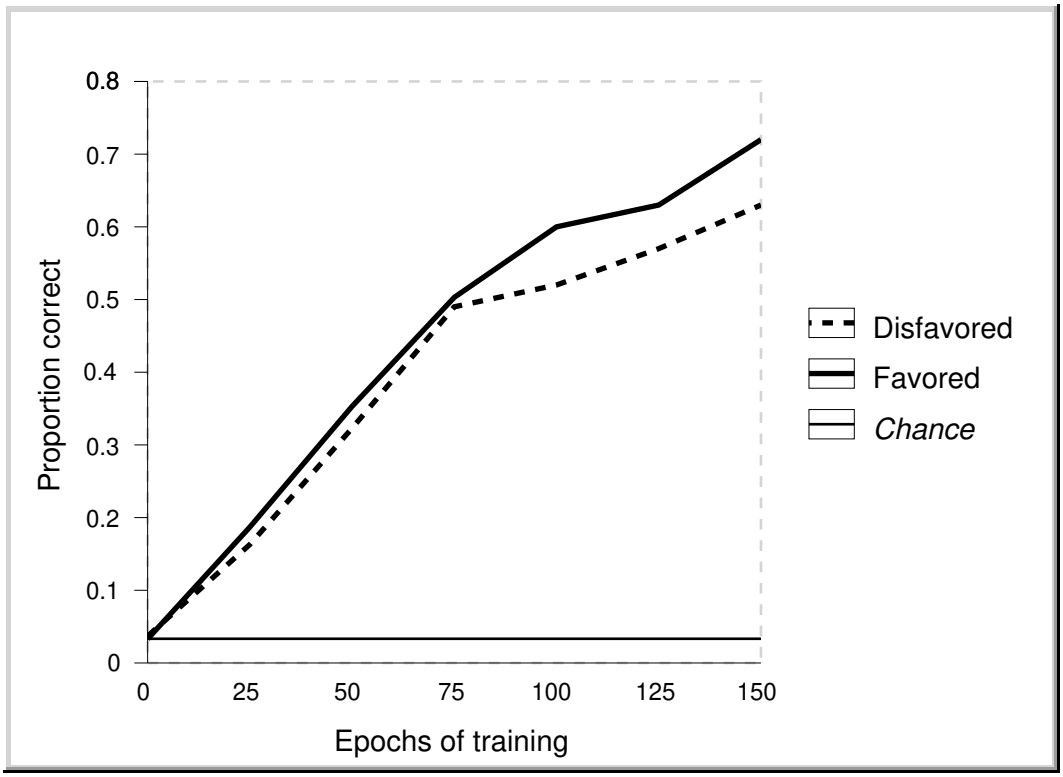

Figure 26: Templatic rules, favored and disfavored, root recognition, modular network

6.7 Morphophonology

In all of the experiments described so far, all roots and inflections have a single form. But, as discussed briefly in Section 2.3, natural languages are not always so simple as this; inflections, and less often roots, can have different forms depending on their environments. However, variation in the form that morphemes take is usually motivated; it causes words to conform to the phonology of the language. A model of the learning of morphology should experience no unusual difficulty learning rules involving morphophonological variation despite the problem of learning to associate multiple forms with a single meaning.

I will consider only one sort of process here, harmony. As with the rest of phonology and morphology, this has been studied almost entirely from the perspective of production, so previous work may not be entirely applicable to the problem at hand. Vowel harmony consists in constraints on the types of vowels which occur within a single word; that is, the vowels must agree on one or more features (roundedness, etc.). In languages with vowel harmony, a particular value for the

relevant feature is normally a property of the root or stem of a word. This means that inflections, if this language has them, may have to have alternate forms to maintain the harmony constraint. This phenomenon plays a significant role in the phonology of highly languages with vowel harmony such as Finnish and Turkish. Consider two hypothetical languages, both of which maintain vowel harmony within stems, but only one of which maintains harmony throughout entire words. If the languages take suffixes which are distinguished by their vowels, we might expect suffix recognition to be somewhat more difficult in the word harmony language because the network cannot simply memorize the vowel in the different suffixes. Root recognition, on the other hand, should be somewhat simpler in the word harmony language because the suffix vowel, as well as the stem, provides information about the root. Thus we would expect a tradeoff in performance on the two tasks.

To test the effects of harmony constraints on the performance of the model, an experiment was conducted in which separate networks were trained on simple suffixing rules, one constrained by harmony, the other not constrained. The roots consisted of 48 CV syllables composed of the consonants /p, b, f, m, t, d, s, n, k, g, x, ng/ and the vowels /i, e, a, o, u/. The present tense of each word consisted of the bare root (stem), while the past tense was formed with the addition of a CV suffix. In the non-harmony case, this suffix was always /ti/. In the harmony case, the suffix consisted of /t/ followed by the vowel of the root. Thus in the latter case, the inflection obeyed a very strict kind of vowel harmony. As before, 2/3 of the forms made up the training set, the remaining 1/3 the test set. Ten separate networks were trained for 100 epochs for each of the two rules, and performance was measured as before. Results for both root and inflection recognition are shown in Figure 27.

Inflection recognition is slightly lower for the harmony rule, as expected. Since the vowel in question is the last segment of the word, in the non-harmony case, the network can solve the task most easily by simply paying attention to this vowel. This strategy does not work in the harmony case, where the vowel varies. Note, however, that in either case, inflection recognition would be a very simple task because all present tense forms have 1 syllable (2 segments) while all past tense forms have 2 syllables (4 segments). In contrast to inflection recognition, root recognition is higher for the harmony rule. Because of the extra information about the root which is provided by the suffix vowel, we see the expected tradeoff. Thus, from the perspective of perception, harmony can be seen as fulfilling a useful function; it facilitates word recognition by adding phonological redundancy.

7 Discussion

7.1 Summary of Results

In this paper, I have described initial investigations into the acquisition of morphology from a new perspective. This perspective is distinguished from most other recent accounts in that it takes seriously three aspects of morphology, (1) the fact that learners are concerned with the mapping of forms onto meanings (and vice versa) rather than forms onto forms, (2) the relationship between perception and production, and (3) the temporal nature of words. I have focused here on the learning of the ability to *recognize* morphologically complex words, which, it is argued, must precede the learning of the ability to produce them. Given the constraint that words are to be presented to the system as sequences rather than all at once, one of the simplest possible devices capable of

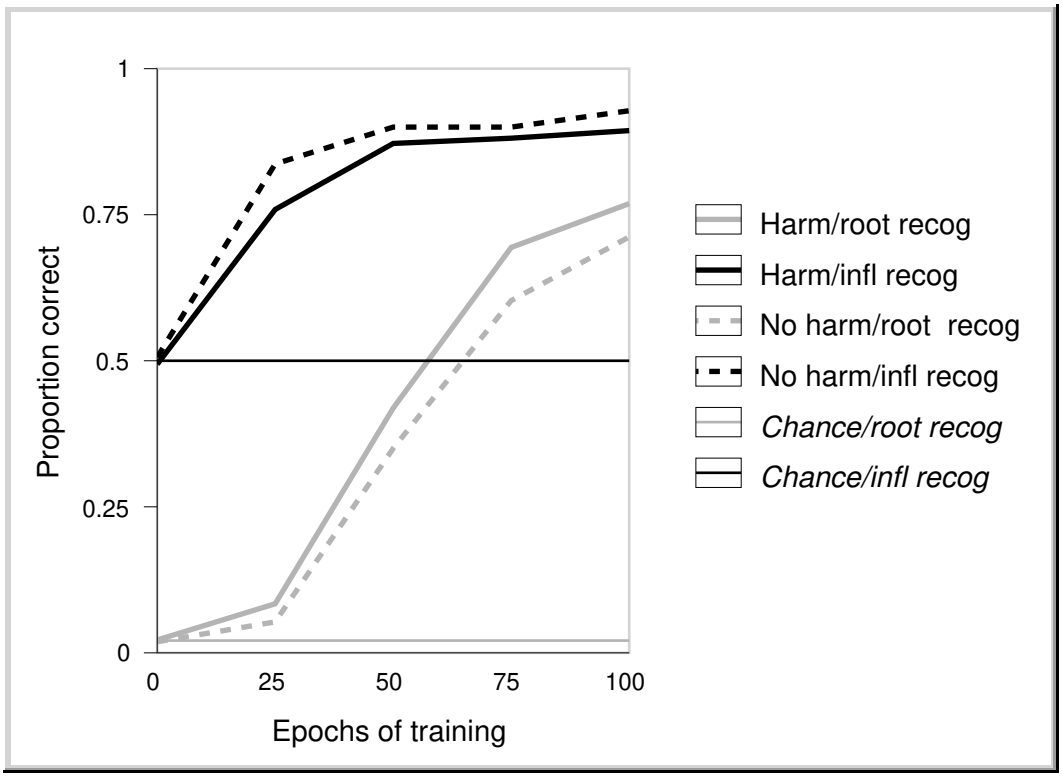

Figure 27: Suffixing rules with and without vowel harmony, modular network

learning morphological rules is a simple recurrent network. The experiments discussed in this paper investigate the capacity of such a network to learn productive morphology for recognition, that is, on the basis of presented word-meaning pairs, to generalize to novel words. As we have seen, this task is really two tasks, the identification of the root (or stem) and the identification of the inflections.

The first conclusion to be drawn is that a simple recurrent network is capable of learning morphological rules for recognition. Only in the case of root recognition for circumfixation rules was performance clearly inadequate. However, the performance of the network is apparently limited by the interference which the two subtasks exert on each other. The capacity of the network is thus improved dramatically when it is outfitted with separate hidden layers for root and inflection recognition. A network which is modularized in this way learns, with different degrees of success, rules of all of the major types found in human languages other than reduplication. There are also initial indications that a network which learns to use the hidden-layer modules it is provided with can settle on an efficient way of sharing the modules among the recognition subtasks. Finally, simple reduplication rules are learned in a network which takes learned distributed representations of syllables as inputs. Further indication that this approach is on the right track comes from experiments demonstrating that rules of a particular type which seem not to occur in natural languages are harder to learn than similar rules which do and that at least one phonological process which may accompany morphological processes does not interfere significantly with performance, and may even improve it.

These experiments have implications in three general areas: the relative difficulty of different categories of morphological rules, modularity in connectionist models of language processing and acquisition, and the role of time in morphology and language in general.

7.2 Factors affecting ease of learning

One of the goals of a theory of language acquisition should be an account of what makes certain forms easier to learn than others. For the learning of morphology, we have seen that there are two components to the task, one involving the root, the other the inflections. And there are no *a priori* reasons to believe that particular types of rules should affect the learning difficulty of the two subtasks in the same manner. There is in fact considerable asymmetry.

A traditional way to break down morphological processes other than compounding is into affixation, mutation, intercalation (templatic rules), deletion, and reduplication. In terms of the difficulty of training a simple recurrent network to learn the rules, we have seen that there is a fundamental division separating reduplication from the other rule types. Reduplication rules, other than the most trivial, are apparently unlearnable by networks which take individual segments as inputs. This is because of the need to compare explicitly whole stretches of segments from the input. As we have seen, however, recognition networks can be trained to represent larger units, such as syllables, and with these representations as inputs, another network can learn to recognize reduplication.

Within the remaining categories of rules, the following factors have been shown to play a role in learning difficulty:

1. Is there a conflict between two aspects of the recognition process? This appears to work against root recognition for circumfixation.

2. Is the preceding context at testing completely different from that seen during training? This works against root recognition for prefixation.

3. Is there a consistent sequence of segments associated with the morpheme being recognized? The effect of this factor is relatively minor; it works against root recognition for infixation, templatic rules, and mutation.

4. For inflection recognition, does the relevant portion of the word occur also in words which are not so inflected? This works in favor of mutation and infixation when the infix or mutated segment is peculiar to the inflection in question.

5. For mutation and infixation, is the position of the infix or mutated segment(s) constant within the word? This works in favor of inflection recognition.

6. Is information about the root unavailable in one or more forms? This works against root recognition in deletion rules.

7. Is there phonological redundancy in the form of an inflection or root? This works in favor of inflection recognition for circumfixation and for some templatic rules and in favor of root recognition when phonological harmony is at work.

8. Does the order of the segments associated with the root vary across the different forms? This works against templatic rules of the "disfavored" type described above.

There are other aspects of rules, not investigated here explicitly, which probably play a role in the performance of a network such as this and which will be examined in future experiments.

1. Various factors related to the confusability of roots are almost certainly related to performance on root recognition. These include the number of phonemes in the language being learned and number of words being learned, both leading to greater confusability.

2. Another set of factors probably affecting root recognition concern the amount of experience the network has with a form during training. Among these would be the number of morphemes within a category (e.g., 3 vs. 2 tenses) and the number of different inflections present on a word (e.g., both tense and aspect vs. only tense).

3. Factors probably affecting inflection recognition include relative consistency in the shape of the inflection, the length of the inflection, variability in the length of the inflection, and consistency in the ordering of inflections when there is more than one.

7.2.1 Modularity

In this paper I have described a model which is modular in three respects. There are separate modules for recognition and production, though they share representations. There are separate modules for recognition of the root and inflection in word recognition. And there are separate modules for processing words at the level of individual segments and at the level of syllables and perhaps higher-level units.

Modularity is a good idea when the two tasks or domains in question interfere with each other, when they place conflicting demands on the resources of the system. This is especially true when the system's resources are otherwise distributed, as they are in the hidden layers of multi-layer perceptrons and their simple recurrent variants, such as the networks examined in this paper, and as they apparently are in much of the brain. Modularity may be built into a system from the start of its development, or it may emerge as the system is exposed to conflicting tasks. The possibility of modularity as an emergent phenomenon in language acquisition has been proposed by Bates, Bretherton, & Snyder (1988), and the adaptive approach to modular networks developed by Jacobs et al. (1991) offers a way to have modularity emerge in a system that can profit from it.

On the other hand, modularity is a bad idea when the tasks or domains in question can benefit from shared hardware, that is, when one stands to generalize on the basis of the other.

In most cases, given two tasks, there will probably be some aspects which conflict and other aspects which are common to the two. This suggests an intermediate possibility: portions of the system dedicated to particular tasks and others available for sharing. The approach of Jacobs et al. (1991) apparently permits this sort of "soft modularity" to emerge in a system where it is called for.

Connectionist networks allow the explicit testing of these various alternatives. Of the three types of modularity discussed in this paper, one, the division between the root and inflection recognition tasks, was directly motivated by an apparent conflict between the two tasks. I showed how the modular arrangement improved performance and also how, under certain circumstances, a network using a modified version of the the algorithm of Jacobs et al. (1991) could learn to use modular hidden layers which had not been pre-assigned to the recognition subtasks. Note, however, that this does not amount to simply setting the algorithm loose on an unorganized system; the algorithm requires that the output tasks already be distinguished. What it does is then decide how or whether

these tasks will use the modules that are available. Thus this approach to the learning of word recognition starts from the following:

1. Each word consists of a fixed set of morphemes. The semantic categories that these come from, including the lexical (root) category, are known beforehand. Recognizing the morpheme for each category, including the root, for a given input word constitutes a separate output task.

2. Modular hidden layers are available so that an efficient way of solving the separate output tasks may be found.

In other words, on the basis of the *semantics*, certain tasks are distinguished. For example, identifying the precise nature of an action (lexical categorization) is distinguished from identifying the time of the action relative to the present (a grammatical categorization). Then the system learns to treat these tasks in separate subnetworks because *phonologically* they require it. The weakness of the approach, in its present form, is that it is not specified how the tasks are distinguished in the first place. In particular, the possible *interaction* between semantic organization on the one hand and morphological/phonological organization on the other is not provided for. It is almost certainly impossible to learn completely the necessary distinctions on the basis of semantics alone because what gets realized grammatically and what lexically differs from language to language. A category such as relative size, for example, takes the form of roots such SMALL and as diminutive markers like those common in Spanish and Russian.

A second type of modularity proposed in this paper involves separate sequential networks for phonological units of different sizes. The situation here is more complicated. In the first case, we started with two given output tasks and modularized the network to suit them. Here the problem is that in order to solve a given output task (the recognition of reduplication), it appears necessary to create an intermediate stage at which there is a new task, one not previously foreseen. This is the segmentation of the input sequence into syllables or other multi-segment units. The original task of mapping segment sequences onto a reduplication morpheme has become two tasks: mapping segment sequences onto syllable sequences and mapping these in turn onto the reduplication morpheme. There are really two issues here. Are syllables or other higher-level units necessary? If we believe in syllables, how does the system deal with them? I have attempted to justify syllable representations on the basis of what is required to recognize reduplication; there have been many other arguments from linguistic and psycholinguistic perspectives (Hogg & McCully, 1987; Cutler & Norris, 1988). If syllables are to be represented, they can either be handled by the same network that handles segments, or they can be treated in a separate sequential module. Modularity of this kind is reasonable again if what the system has to learn about sequences of syllables has little in common with what it has to learn about sequences of phonemes. This type of modularity is to be contrasted with what Hinton calls **between-level sharing** (Hinton, 1990), which is called for in domains, such as vision, in which the same general knowledge applies to different levels. For language modularity may make sense. This leaves the problem of what sort of system could organize itself in this fashion. At present I have no more than the vaguest ideas about this. In any case, such a system would probably need considerable pre-wiring.

The third kind of modularity proposed here, that dividing word recognition from word production, is also not particularly controversial. At the periphery, these two processes involve systems which are clearly distinct, audition and articulation. At other end, they apparently share semantics. An extreme modular position would leave it at that. Yet, as I argued above, this leaves production

acquisition to fend for itself in a way which seems highly implausible. With sharing between perception and production at some intermediate phonological level, production can benefit from what perception learns. An appealing idea is that the intermediate representations that appear to be called for for reduplication recognition also serve as the representations which mediate the mapping from semantics to articulation. Notice that this creates a fourth modularity, similar to the second: production now consists of two subtasks, the mapping of semantics onto sequences of intermediate phonological representations and the mapping of these representations onto sequences of articulatory gestures. A major challenge is the design of a system which can *learn* what it is that perception and production share.

7.2.2 Time and Language

The approach described in this paper differs from most approaches to morphology or, for that matter, to language, in that the phenomena under investigation take place in time. That is to say, the system is never given direct access to all of the elements in the sequences it is classifying. Are there aspects of natural language morphology which are a reflection of the fact that language happens in time, or is this fact peripheral to all of the processes that have concerned us in this paper?

The left-to-rightness of words leads to a number of asymmetries in the performance of the network. Prefixation and suffixation behave differently because, from the network's perspective, it is the previous context of the affix in both cases that, together with the affix itself, determines performance. Previous context for a prefix means the edge of the word, whereas this is not true for a suffix. It is also the sequential nature of the processing that makes the "disfavored" templatic rule harder than the "favored" rule. Because the nature of the representations developed by the network depends crucially on the order in which the segments are processed, the order matters: *garam* will not look like *gamar*.

But there is also a sense in which language seems to be composed of static units; the sequences which are words need eventually to map onto the meanings of the morphemes making up the words, and concepts such as CUP and PLURAL do not seem to be sequences. Thus word recognition is in part about mapping sequences onto static units. But if there are intermediate stages in this process, as there seem to need to be, at least in order to accomplish the recognition of reduplication, and these stages are also phonological, then they would also be sequential. So the process is one of mapping sequences of smaller units onto sequences of larger units and eventually to static entities. This is a version of the traditional hierarchical view of language which accords a role to time. In this paper I have shown how the hidden layer patterns of simple recurrent networks can provide the link between the levels in such a hierarchy; following a sequence of inputs, the hidden layer constitutes a representation of that sequence and may serve as a single input to a higher-level sequence processor. Of course, many aspects of this picture remain unclear. How is the segmentation performed? How does the system *learn* to segment? Is a strict separation between levels, say, phoneme and syllable, necessary, or is some form of soft modularity more appropriate? These questions are guiding current work on the model.

7.3 Limitations and Future Work

I have discussed some of the limitations of the model which relate to the self-organization of the network into modules of the type being proposed. I noted there how the strict division of labor into a morphological/phonological component and a semantic component (whose workings are not

explained here) is problematic because it does not permit the interaction that seems to be required to distinguish the lexical and grammatical categories within the target language. But there is much else that the model needs to do before it can be compared to some of the full-fledged symbolic models of the acquisition of morphology such as MacWhinney (1978) and Pinker (1984), hence the "towards" in the title of the paper.

Some of what needs to be accomplished is at least testable in reasonably straightforward ways within the current framework. These include the learning of words containing more than one lexical morpheme; the learning of multiple exponence, that is, the signaling of features from more than one semantic category by a single morpheme; the learning of more complex reduplication processes; the accommodation of multiple rules, including irregular rules, within the same network; the accommodation of words of different syntactic categories, say, nouns and verbs, within a single network; the effects of factors such as inflection length and number of morphemes within a morphological category; the use of semantic micro-features in place of localized morpheme units; and adaptive modularization with varying numbers and types of initial modules and varying types of rules to be learned.

Other extensions of the model are not so simple to accommodate. I will consider just three here: the learning of gender systems, interactions among phonological processes, and segmentation of inputs.

Gender systems classify lexical items into groups, often on the basis of their shape, but only partially, if at all, on semantic grounds. Gender figures in agreement among words bearing particular grammatical relations to one another. Thus children learning French must eventually figure out that *verre* 'glass' is masculine and *tasse* 'cup' feminine so that they can use the appropriate form of an adjective (e.g, *petit* 'little (masculine)', *petite* 'little (feminine)') or pronoun (*il* 'it (masculine)', *elle* 'it (feminine)') to refer to the objects. Gupta & MacWhinney (1992) describe a connectionist model designed to learn complex gender systems which incorporates an explicit memory of co-occurrences and some hard-wiring specific to the task at hand. It is of interest to establish whether the present framework permits the learning of gender without such augmentations. Because no semantic features characterize the gender distinction, there is no way for the model to learn gender morphology directly as it does other morphemes. Of course, because gender in a language like French has essentially nothing to do with semantics, it has relatively little significance for a comprehension system such as the one being modeled here, but it does matter for production, and since we are assuming that production builds on recognition, the distinction must somehow be acquired as a part of the process we are modeling here. For a language such as French, where the nouns themselves generally have no overt indication of their gender, the learning of gender would require whole noun phrases to be presented to the network rather than individual nouns. Noun phrase recognition would be treated as if it were the recognition of a single word containing possibly more than one lexical morpheme as well as possible grammatical morphemes. Given this modification, there is at least the possibility that the network would come to cluster nouns of different genders on the hidden layer of the network on the basis of their co-occurrence with particular input forms. For complex systems such as the German one, however, this seems unlikely to suffice, though some form of defensible incremental training is likely to help.

Morphophonology concerns patterns of phonological variation that depend on the morphological processes involved. I have discussed two simple examples of how the model responds to a rule incorporating morphophonology, and other work has shown that networks have the capacity to learn simple phonological rules in the production direction (Gasser & Lee, 1990; Hare, Corina,

& Cottrell, 1990; Hare, 1990). There is much more to morphophonology than this, however. Of particular interest are cases in which a number of different interacting phonological processes are conditioned by the combination of morphemes. In connectionist as well as symbolic models, these processes have generally been approached from the perspective of generating surface forms given underlying representations (Goldsmith, 1992; Touretzky & Wheeler, 1990). Thus the current model would see the processes in a very different light, most importantly, from the standpoint of perception rather than production. Anyone familiar with the intricacies of the sorts of interactions that can take place, however, would have strong doubts about the efficacy of the simple networks proposed here to handle multiple rules. In particular, there are many arguments in favor of at least three separate levels within and between which phonological processes take place. Perhaps the idea of separate subnetworks responsible for phonological units of different sizes, proposed here to deal with reduplication, could help to solve this problem. Note, however, that this sort of modularity would not correspond directly to the levels of other connectionist phonological models, which are distinguished in terms of their degree of abstraction away from the surface rather than the size of the units involved.

In the present model, inputs to the network are pre-segmented: each word is preceded and followed by a boundary indicator; that is, the word has been separated from the stream of words in which it might have occurred. This simplification may be justified somewhat because words, nouns in particular, which by themselves do not make up sentences, do occur in isolation, and in languages with extensive inflectional morphology, a single word, normally a verb, often constitutes a sentence. This says nothing about how segmentation takes place for input consisting of multi-word utterances though. It has been shown that simple recurrent networks are capable of a sort of implicit segmentation (Doutriaux & Zipser, 1990; Elman, 1990), and in the present model, the network often has to perform a rudimentary sort of segmentation into morphemes within the word. But it remains to be seen whether this mechanism would be powerful enough to deal with multiple words. Again some plausible sort of incremental training can probably help, including early training on simple word spotting tasks.

Another form of pre-segmentation has already broken words into the constituent phonetic segments which are the inputs to the system. Segmentation at this level may be accomplished by separate special-purpose mechanisms which turn an input wave form into sequences of discrete patterns of one type or another (phonetic segments, syllables, etc.).

8 Conclusion

Language is a complex phenomenon. Even a cursory familiarity with a range of languages is enough to convince one that simple solutions will not suffice. The strategy adopted in this paper has been to start with the simplest mechanisms that might be capable of doing the job, as I have defined the job, and then augment the approach when these mechanisms don't suffice. Not surprisingly, augmentations were called for, even for the relatively narrow range of phenomena examined in this paper. These constituted two forms of modularity: for the subtasks of root and inflection recognition and for phonetic/phonological units of different sizes. These modifications are more than convenient hacks for solving technical problems, though; each is motivated on independent grounds. The first is the result of a very general mechanism which solves cognitive tasks by assigning known subtasks to separate modules, where this is to the system's advantage. The second, while not, at least in the current version of the model, something that would arise from an adaptive modularity approach,

agrees with much current work in phonology and psycholinguistics in treating phonemes, syllables, and metrical feet as real in some sense and as fundamentally different from one another.

Can connectionist networks which are more than uninteresting implementations of symbolic models learn to generalize about morphological rules of different types? Much remains to be done before this question can be answered, but, for recognition at least, the tentative answer is yes.

References

Bates, E., Bretherton, I., & Snyder, L. (1988). *From First Words to Grammar: Individual Differences and Dissociable Mechanisms*. Cambridge University Press, Cambridge.

Berko, J. (1958). The child's learning of English morphology. *Word*, *14*, 150–177.

Chater, N. & Conkey, P. (1992). Finding linguistic structure with recurrent neural networks. *Annual Conference of the Cognitive Science Society*, *14*, 402–407.

Corina, D. P. (1991). *Towards an Understanding of the Syllable: Evidence from Linguistic, Psychological, and Connectionist Investigations of Syllable Structure*. Ph.D. thesis, University of California, San Diego.

Cottrell, G. W. & Plunkett, K. (1991). Learning the past tense in a recurrent network: acquiring the mapping from meaning to sounds. *Annual Conference of the Cognitive Science Society*, *13*, 328–333.

Cutler, A. & Norris, D. (1988). The role of strong syllables in segmentation in lexical access. *Journal of Experimental Psychology: Human Perception and Performance*, *14*, 113–121.

Daugherty, K. & Seidenberg, M. (1992). Rules or connections? the past tense revisited. *Annual Conference of the Cognitive Science Society*, *14*, 259–264.

Dell, G. S. (1986). A spreading activation theory of retrieval in sentence production. *Psychological Review*, *93*, 283–321.

Doutriaux, A. & Zipser, D. (1990). Unsupervised discovery of speech segments using recurrent networks. In Touretzky, D., Elman, J., Sejnowski, T., & Hinton, G. (Eds.), *Proceedings of the 1990 Connectionist Models Summer School*, pp. 303–309. Morgan Kaufmann, San Mateo, CA.

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

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

Fahlman, S. E. (1989). Faster-learning variations on back-propagation: an empirical study. In Touretzky, D. S., Hinton, G., & Sejnowski, T. (Eds.), *Proceedings of the 1988 Connectionist Models Summer School*, pp. 38–51. Morgan Kaufmann, San Mateo, California.

Gasser, M. & Lee, C.-D. (1990). Networks that learn about phonological feature persistence. *Connection Science*, *2*(4), 265–278.

Gasser, M. & Lee, C.-D. (1991). A short term memory architecture for the learning of morphophonemic rules. In Lippmann, R. P., Moody, J. E., & Touretzky, D. S. (Eds.), *Advances in Neural Information Processing Systems 3*, pp. 605–611. Morgan Kaufmann, San Mateo, CA.

Gasser, M. & Smith, L. B. (1993). Learning noun and adjective meanings: a connectionist account. Tech. rep. 382, Indiana University, Computer Science Department, Bloomington, IN.

Gasser, M. (1992). Learning distributed syllable representations. *Annual Conference of the Cognitive Science Society*, *14*, 396–401.

Goldsmith, J. (1992). Local modeling in phonology. In Davis, S. (Ed.), *Connectionism: Theory and Practice*, pp. 229–246. Oxford University Press, New York.

Gupta, P. & MacWhinney, B. (1992). Integrating category acquisition with inflectional marking: a model of the german nominal system. *Annual Conference of the Cognitive Science Society*, *14*, 253–258.

Hare, M. & Elman, J. (1992). A connectionist account of English inflectional morphology: evidence from language change. *Annual Conference of the Cognitive Science Society*, *14*, 265–270.

Hare, M., Corina, D., & Cottrell, G. W. (1990). A connectionist perspective on prosodic structure. In *Proccedings of the Annual Meeting of the Berkeley Linguistic Society, 15*.

Hare, M. L. (1990). The role of similarity in Hungarian vowel harmony: a connectionist account. *Connection Science*, *2*.

Hertz, J., Krogh, A., & Palmer, R. (1991). *Introduction to the Theory of Neural Computation*. Addison-Wesley, Reading, MA.

Hinton, G. (1990). Mapping part-whole hierarchies into connectionist networks. *Artificial Intelligence*, *46*, 47–75.

Hoeffner, J. (1992). Are rules a thing of the past? the acquisition of verbal morphology by an attractor network. *Annual Conference of the Cognitive Science Society*, *14*, 861–866.

Hogg, R. & McCully, C. B. (1987). *Metrical Phonology: A Coursebook*. Cambridge University Press, Cambridge.

Jacobs, R. A., Jordan, M. I., & Barto, A. G. (1991). Task decomposition through competition in a modular connectionist architecture: the what and where vision tasks. *Cognitive Science*, *15*, 219–250.

Jordan, M. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, pp. 531–546 Hillsdale, New Jersey. Lawrence Erlbaum Associates.

Kempen, G. & Hoenkamp, E. (1987). An incremental procedural grammar for sentence formulation. *Cognitive Science*, *11*, 201–258.

Kim, J. J., Pinker, S., Prince, A., & Prasada, S. (1991). Why no mere mortal has ever flown out to center field. *Cognitive Science*, *15*, 173–218.

Lee, C.-D. (1991). *Learning to Perceive and Produce Words in Connectionist Networks*. Ph.D. thesis, Indiana University, Bloomington.

MacWhinney, B. & Leinbach, J. (1991). Implementations are not conceptualization: revising the verb learning model. *Cognition*, *40*, 121–157.

MacWhinney, B. (1978). The acquisition of morphophonology. *Monographs of the Society for Research in Child Development*, *43*(1–2, Serial No. 174).

Marcus, G. F., Pinker, S., Ullman, M., Hollander, M., Rosen, T. J., & Xu, F. (1992). Overregularization in language acquisition. *Monographs of the Society for Research in Child Development*, *57*(4, Serial No. 228).

Marcus, G. F., Brinkmann, U., Clahsen, H., Wiese, R., Woest, A., & Pinker, S. (1993). German inflection: the exception that proves the rule. Tech. rep. Occasional Paper #47, MIT Center for Cognitive Science, Cambridge, MA.

Markman, E. M. (1989). *Categorization and Naming in Children: Problems of Induction*. MIT Press, Cambridge, MA.

Marslen-Wilson, W. D. & Tyler, L. K. (1980). The temporal structure of spoken language understanding. *Cognition*, *8*, 1–71.

Martin, J. (1988). Subtractive morphology as dissociation. In *Proceedings of the Seventh West Coast Conference of Formal Linguistics* Stanford, CA. Stanford Linguistics Association.

Mtenje, A. (1987). Tone shift principles in the CHICHEŴA verb: a case for a tone lexicon. *Lingua*, *72*, 169–209.

Norris, D. (1990). A dynamic-net model of human speech recognition. In Altmann, G. T. M. (Ed.), *Cognitive Models of Speech Processing: Psycholinguistic and Computational Perspectives*, pp. 87–104. MIT Press, Cambridge, MA.

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

Pinker, S. (1984). *Language Learnability and Language Development*. Harvard University Press, Cambridge, MA.

Plunkett, K. & Marchman, V. (1991). U-shaped learning and frequency effects in a multi-layered perceptron: implications for child language acquisition. *Cognition*, *38*, 1–60.

Port, R. (1990). Representation and recognition of temporal patterns. *Connection Science*, *2*, 151–176.

Pullum, G. (1982). Letter. *Linguistics*, *20*, 339–344.

Regier, T. (1992). *The Acquisition of Lexical Semantics for Spatial Terms: A Connectionist Model of Perceptual Categorization*. Ph.D. thesis, University of California, Berkeley.

Rumelhart, D. E. & McClelland, J. L. (1986). On learning the past tense of English verbs. In McClelland, J. L. & Rumelhart, D. E. (Eds.), *Parallel Distributed Processing, Volume 2*, pp. 216–271. MIT Press, Cambridge, MA.

Selkirk, E. O. (1982). The syllable. In van der Hulst, H. & Smith, N. (Eds.), *The Structure of Phonological Representations, Part II*. Foris, Dordrecht.

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

Spencer, A. (1991). *Morphological Theory*. Basil Blackwell, Oxford.

Stevens, A. (1968). *Madurese Phonology and Morphology*. American Oriental Society, New Haven, CT.

Touretzky, D. & Wheeler, D. (1990). A computational basis for phonology. In Touretzky, D. (Ed.), *Advances in Neural Information Processing Systems 2* San Mateo, CA. IEEE, Morgan Kaufmann.

Waibel, A. (1989). Modular construction of time-delay neural networks for speech recognition. *Neural Computation*, *1*, 39–46.