

# Learning inflectional classes

John Goldsmith and Jeremy O'Brien  
University of Chicago

March 10, 2006

## 1 Introduction

We begin by providing an overall context to the problem that we consider in this paper. The language learner is faced with the task of learning phonology, morphology, syntax, and semantics. Our familiarity with generative grammar in recent years has tended to emphasize what linguists call “syntagmatic” relations, which is to say, how the various morphemes and words of any given utterance are related to each other, in morphological or in syntactic ways: if *child* is the subject of *laughed* in *the child laughed*, this is a *syntagmatic* relationship. But the grammatical structure of a *language* (as opposed to an utterance) includes other structure as well, and this other structure must also be learned: this is the *paradigmatic* structure, which is particularly important in the description of a language’s morphology. The term *paradigmatic structure* refers to the way in which different choices for lexical or grammatical realization at any given spot in an utterance are organized.

A simple case of paradigmatic structure is the possibility of singular and plural forms for many nouns in English. Thus the task of learning English will include: (i) learning that most nouns have a singular and a plural, though some do not have both (some have no plural, or only a plural to indicate different types, such as *milk* or *wine*; some have no singular (*scissors*, *pants*); and the complexities of English do not end there, of course); (ii) learning how to form the plural form of a noun (most nouns take a suffix */-z/*, which then may undergo various phonological modifications, but some nouns take irregular suffixes (*children*, *oxen*) including null suffixation (*fish*, *sheep*)); (iii) learning that this binary distinction plays a role in the agreement phenomena linking a subject with an inflected verb: such verbs must “agree in number” with their subject.

While the singular/plural distinction is no doubt a prototypical example of a paradigmatic opposition, it has two characteristics that are often not shared by paradigmatic oppositions in morphology: first, the singular/plural distinction corresponds rather well to a semantic distinction, and second, it often is involved in an overt agreement process involving a different word in the sentence (in the case at hand, the agreement of the verb to its subject). Some morphological paradigmatic oppositions do not express a semantic distinction, and some do not trigger syntactic agreement rules. The aspectual choice of a main clause verb, for example, may indicate whether the action being described is completed or not, but this choice may have no formal consequences in a sentence beyond the choice of form of the verb itself (this case is illustrated

by the difference between the Spanish imperfect *cantaba* ‘(s)he was singing’ vs simple past *cantó* ‘(s)he sang’). Still, such a distinction lines up well with one or more semantic oppositions. But that is not always the case. There exist morphological oppositions that carry no semantic weight, and do not play a role in any agreement phenomena—but which need to be learned nonetheless by the language learner. The prime example of such paradigmatic oppositions is found in what are traditionally called *inflectional classes*. Anyone who has had to learn the various groups of verbs in French, Spanish, or Latin in school has encountered such groupings of verbs, and children even younger are often presented with the challenge of learning these patterns in the ambient language available to them. That challenge is the general subject of this paper: the task of learning morphological categories that are aligned neither with meaning nor with syntax.

The last ten years have seen a great deal of work on computational approaches to the learning of natural language morphology. Much of that work has focused on the problem of *segmentation* of words into their component morphemes (e.g., [6], [12], [18]; some of it has aimed at inducing other aspects of morphological structure, such as morphosyntactic features. The study of word and morpheme segmentation is one where both the computational literature (just cited) and the psycholinguistic literature (Aslin Newport Saffran etc) has independently moved towards the utilization of probabilistic models and techniques. As we have noted, the present paper addresses a different (though not unrelated) question, that of the clustering of alternative forms of affixes into the groups that linguists have traditionally referred to as inflectional classes (using also the terms *declensional* class for nouns and adjectives, and *conjugational* class for verbs). In so doing, it returns to a subject discussed by MacWhinney et al. [25], and we will discuss our results and conclusions in relationship to theirs below.

## 2 What kind of generalizations are inflectional classes?

The inflectional classes of verbs in English (and more generally, the Germanic languages) are constituted by what are traditionally called the *weak* and the various *strong* classes: weak verbs (also misleadingly called ‘regular verbs’) form their past with the suffix *-ed*, while strong verbs form their past with the suffix *t*, and internal vowel change, or both. Verbs of the Romance languages, including Latin (and many other language families as well), are generally divided into inflectional groups identified by the choice of the infinitival suffix, such as *er* verbs, *ar* verbs, etc. As the name suggests, inflectional classes classify the *way* in which categories of inflectional morphology are realized on a given verb in a given language.

In many languages—especially Indo-European languages, but not only them—a large proportion of the inflectional morphology involves morphemes which realize conjunctions of morphosyntactic features, such as the *-o* suffix in Spanish which realizes the conjunction of 1st person, singular, and present tense. This characteristic lies at the heart of the nearly superannuated term “inflectional language.” The traditional way to render this system perspicuously is to place

the morphemes in a two-dimensional tableau, or in a higher-dimensional tableau when appropriate, though the graphical presentation is typically broken down into a series of two-dimensional presentations, as in Table 1.

It is also common to find that two or more morphemes realize the same conjunction of morphosyntactic features, as *-amos*, *-emos*, and *-imos* are all realizations of the 1st person plural present tense verbal suffix in Spanish, with each stem selecting one and only one of these forms. But it has been known since antiquity that stems do not choose *randomly* from the choices of realizations of morphosyntactic features; rather, they fall into a small number of patterns of affix-selection that have traditionally been called conjugations (for verbs) and declensions (for nouns)—or more generally, inflection. (A third common characteristic of these systems is what is known as *syncretism*: the apparent use of the same morph (that is, the same set of sounds) in a number of distinct sets of the morphosyntactic features.)

1st conjugation	Singular subject	Plural subject
1st person subject	-o	-amos
2nd person subject	-as	-áis
3rd person subject	-a	-an

2nd conjugation	Singular subject	Plural subject
1st person subject	-o	-emos
2nd person subject	-es	-éis
3rd person subject	-e	-en

3rd conjugation	Singular subject	Plural subject
1st person subject	-o	-imos
2nd person subject	-es	-ís
3rd person subject	-e	-en

Table 1: Traditional tableau of Spanish present tense suffixes

Aronoff [2] describes, and defends, a linguistic perspective that asserts the existence of morphology as a component that exists for and by itself (and it is a view which to some degree we will defend as well, though not its characterization of tableaux). He employs an image of morphology as including tables much like those in Table 1. If we chose one of the columns and one of the rows, we have what Aronoff calls a *morphosyntactic property array*, and he says:

According to my definition, a language will have inflectional classes when there is more than one realization for a given morphosyntactic property array. In Latin, for example, where case and number are always jointly realized, there is almost always more than one suffix by which a particular case and number array may be realized. For a given noun or adjective, we must therefore know, for each case and number array, which of its realizations will appear.

(64)

As the reader can see, Table 1 illustrates the sort of case that Aronoff is referring to: since there are three different ways to realize the combination *first*

*person* and *plural*, the language has inflectional classes. Paraphrasing Aronoff now, we may say that if we fix a particular stem, then we can think of the morphology as a device that is able to produce (or generate) all the necessary forms of that stem as we vary the “input” morphosyntactic features (like case and number, for example). For a given part of speech, an inflectional class is the set of all affixes that are produced by that device (where we can think of each affix as being tagged by the features that caused it to be output). If all the stems of a language are associated with the same inflectional class, then life is simple; but when indeed *not* all stems produce the same set of suffixes, we say that there is more than one inflectional class in the language. A moment’s thought will show that this will happen if, and only if, there is at least one set of input features that produces more than one output affix, depending on which stem is chosen. Now, in theory at least, if there are  $K$  different combinations of input features that each have at least two different affixes that they can generate, we would be within our rights to expect  $2^K$  different inflectional classes. However, that expectation is not fulfilled; Aronoff, citing Carstairs [4], notes that “normally the total number of classes is minimal: it is equal to the highest number of choices of phonological realizations available for a single morphosyntactic property array”.

The business of the lexicon is, among other things, to ensure that each stem knows which set of affixes it should be associated with. Returning to Aronoff:

The lexical entry for the noun must therefore bear some sort of flag to assure that it will manifest the appropriate set of inflections. This flag is the inflectional class of the noun. Membership in a given inflectional class will guarantee that the noun has exactly the realization pairs of that class... Mechanically, as a way of insuring that all the realization pairs of a given class will be linked to members of that class, I add the class name to the first member of each realization pair. The class thus acts as a rule feature or rule trigger, and the realizations characteristic of a given class or paradigm will all be conditioned by that class name or class flag or rule trigger. *[There is a] usually unmotivated, abstract morphological category, the inflectional class, which determines not the stems but rather the inflections of lexemes.* I would like to emphasize again that what I have said about gender and inflectional class is truly ancient. It was not for nothing that the Latin grammarians only gave numbers to inflectional classes for both nominals and verbs. The numbers are arbitrary, which shows the arbitrary basis of the inflectional classes. That linguists, in their quest for motivation, still ignore this important finding despite its frequent repetition may tell us something about ideology, but it should not gainsay the truth of the original observation: that Latin inflectional classes are demonstrably divorced from gender and singularly unmotivated. (p. 167; emphasis added)

It would perhaps not be an exaggeration to say that current morphological theory gives us no clue as to how the learner of a language learns that there *are* such paradigms, and how a speaker can generalize *from* hearing a small number of forms of a given stem *to* a conclusion that the other unheard forms must be like some pattern that has already been extracted from the data.

But *views* on the subject are not in short supply, and they may be roughly categorized into three groups, in increasing order of the sophistication they assume in the learning abilities of language acquirers:

1. Innatist position: Universal grammar allows for a number of paradigms—that is, alternative realization patterns of surface morphemes—for the major stem classes of natural languages, such as nouns, verbs, adjectives, and the like. Learners use the existence of these alternatives to organize—that is, to make sense out of—the apparently inconsistent way in which grammatical features like “plural” and “1 person plural” are realized in the language that they are learning.

2. Word-exemplars: paradigms exist because learners learn complete paradigms of certain words—perhaps especially high frequency words. Learners then generalize these forms to other stems. This perspective is well characterized by Martin Haspelmath [21] in a recent textbook; we will return to his comments shortly.

3. Emergence: In MacWhinney et al.’s [25] words, “paradigms emerge on the basis of associations between cues” (p. 274), and they interpret this account as being one that “establishes no formal classes apart from the patterns of associations within the [connectionist] network” that they model.

Our perspective falls into the third group, and in some respects, shares certain elements of that of MacWhinney et al. That is to say, we agree that it is a good result that purely formal, or mechanical, manipulation of linguistic data from a natural language can lead to an analysis which can then be put in a point-by-point relationship with traditional linguistic analyses, and to the extent that these purely formal means do not (either implicitly or explicitly) build in the notion of paradigm, it is fair to say that we have achieved a level of linguistic explanation that we may not have had before, and that we would not have had if we had, following an innatist perspective, merely built the answer into the learning model. On the other hand, to say that a connectionist model offers some insight into the presence of inflectional classes in language is not to say that the generalizations are not *morphological*; grammar needs its own proper organization regardless of whether we find particular approaches to learning to be useful (or not). To put the same point another way, it is our view that developing a connectionist model of a linguistic phenomenon is not tantamount to eliminating the need for a grammatical analysis.

MacWhinney et al. are at pains to associate their results with a perspective on language in which formal classes are suspect, and in which language use, like language acquisition, does not surpass the abilities of simple connectionist models. It seems to us that is not a viable option in light of our current knowledge. What we do understand about morphology induction ([6] [12] [18]) establishes firmly, in our view, that the learning of a natural language morphology must include a remarkably complex and abstract optimization of grammar simplicity and adequacy in the treatment of data. The task of segmenting speech into words ([3], [7]), and words into morphemes ([17]), is a complex one in which probabilistic features are essential to the model, and an adequate model incorporating probabilities bears no resemblance whatsoever to connectionist models (we return to this point immediately).

In our view, traditional grammatical theory (which is still in flux and development, a state of affairs that will not go away) has evolved over time in order to allow traditional linguists the vocabulary with which to *describe* the

complexities of natural language, an extremely difficult goal to reach, and it has not been specifically designed with the aim in mind of accounting for acquisition. For better or worse, it turns out to be extremely difficult to figure out how to embed traditional grammatical theory within a broader framework that can account for learning. The connectionist models that MacWhinney et al. explored are, by contrast, exquisitely designed for learning some (interesting) kinds of classification problems, but are of little use for tasks for which they were not designed, such as the organization of grammatical information in natural languages.<sup>1</sup>

Our goal, then, is to see to what extent we can learn from the piecemeal importation of learning models (whether based on neural networks, bayesian models, hidden Markov models, support vector machines, minimum description length models, or any others) into our understanding of linguistic description. We will return to the broader questions below in Section 6, but we will say upfront that it is not, in our view, a desideratum that we not find traditional grammatical categories in our model of language learning, and it is certainly very possible that we may have to do some work to discover the placement and location of traditional grammatical categories in our learning models.

Returning to the word-exemplar view noted above, we were struck by the following passage from a recent textbook on morphology, by Martin Haspelmath [21]. The reader will note the allusion in the first sentence to the kind of analysis, discussed above, which Aronoff [2] was at pains to defend.

One traditional approach [to the problem we are discussing] consists in giving arbitrary diacritic names to the paradigm rules that describe the inflection classes, and marking each lexeme with a **diacritic** feature in its lexical entry. In Latin, there is a tradition of numbering the declension classes that goes back to the grammarian Priscian. In this tradition, *hortus* would be said to belong to the second declension, and *gradus* to the fourth declension. For example, the lexical entry of *hortus* would contain, in addition to its phonological, syntactic and semantic features, a purely morphological diacritic feature ‘[second declension]’. Thus this approach requires a special look-up procedure that creates the correct inflected forms

---

<sup>1</sup>This is not the place for a detailed critique of the range of connectionist models of language, and obviously we are sympathetic to some aspects of connectionist modeling of learning, but we will mention briefly three reasons why we think that connectionist modeling of language still have a lot of work ahead of them in becoming full-fledged models of grammatical structure. Interesting efforts made in the late 1980s to deal with syntactic constituent structure (of which the best were Smolensky’s tensor-based model ([34], reviewed in [35]) and Pollack’s RAAM model ([32], [22])); they remain to be worked out. In addition, the treatment of temporal order in connectionist models is problematic. The curious fact is that the future often influences the past, when it comes to language: for example, many languages declare that the even-numbered syllables of a word, *counting from the end*, are stressed. Such a generalization can only be modeled with a connectionist system in which space plays the role of time: which is to say, in which distinct connectionist units are used to model each syllable of a word, as in [14], [15], [19]. Such a move would, it is fair to say, be accepted by no one for syntactic structure; one of the central hypotheses of connectionist modeling is that the dynamics of language utterance can be reduced to the dynamics of the connectionist models themselves. A third serious drawback to connectionist modeling is the difficulty of using the insights of bayesian language modeling; that is, it is difficult to conceive of a connectionist-style learning algorithm that assigns a prior probability to long-term weights that corresponds to the inverse of the notion of grammar complexity.

on the basis of the paradigm rules that are stored in the grammar. This is a very economical solution in some sense, because numerical diacritics are easy to handle, and many dictionaries of languages with global inflection classes have adopted it. However, it is questionable whether the human cognitive apparatus allows the device of diacritic features for inflection classes. Diacritic features are very efficient in computer programs, but there does not seem to be any other evidence that human memory ever makes use of such arbitrary diacritics.

Thus, linguists who are interested in cognitively realistic descriptions of inflection have looked for alternatives. A possibility that has often been adopted by dictionaries as well is to list several **reference forms** of the lexeme that together permit a unique assignment to one of the inflection classes. For example, descriptions of Latin often give the genitive singular form of a noun in addition to the nominative (*hortus, hortī; gradus, gradūs*). Since there are no two declension classes [in Latin] that have the same suffixes in the nominative and genitive singular, this method provides the same information as the declension class diacritic. And, unlike the diacritic, it can plausibly be applied to models of the mental lexicon. Speakers can be assumed to remember the nominative and genitive singular forms of each lexeme, and, by matching these two forms against the word-schemas in their paradigm rules, they can determine all the other forms of the paradigm. Remembering two inflected forms instead of a stem and a diacritic feature is less economical, but probably more realistic. (123-24) [21].

At the risk of caricaturing his point, it seems to us that Haspelmath is saying that if someone walks down the street and looks at the passersby, she may say, That person looks just like my Aunt Mabel, and that one's eyes and forehead reminds me ever so much of my student Svetlana. But (if we understand Haspelmath's view) she would not think, That person looks very Swedish, and that one looks like an Ethiopian—because making such generalizations, akin to inflectional paradigms, is not a cognitively realistic description of what people can do. We do not share this hunch with Haspelmath, and think that people are indeed endowed with the ability to generalize to form categories that involve the complex integration of data from a large number of experiences.

We view the linguistic problem differently, and propose to explore this alternative in an explicit computational system, one that employs neural network learning techniques. We believe that linguists and psychologists must work jointly to understand what is a likely model of human learning and generalization, and in particular we are aware (*pace* Haspelmath) of no psychological evidence that the traditional notion of the paradigm embodies a concept that is difficult for humans to learn. In fact, we take a rather different tack here; we ask what the characteristics of a learning device might be like if it is lead to view linguistic data in terms of inflectional paradigms? While we cannot perhaps make the very strong assertion that the organization of verbal and nominal data into paradigms is not inherent in the data, or that the data does not wear its organization on its sleeve, it is nonetheless certainly not obvious how a learning device would be lead from the data to organize it in such a

fashion. We explore the bounded problem of a device which takes as its input a stem and a set of morphosyntactic features, such as 1st person, singular, past, and produces a *suffix choice* as its output.<sup>2</sup> Furthermore, we expect the device to be one which is amenable to *learning*: that is, we will present the device with a finite number of cases in which a set of (correct, or perhaps mostly correct) input stems-and-features are presented along with a single suffix choice in the output. This finite number of cases is the training data, and we then expect to be able to feed various combinations of stems and morphosyntactic features (both from within the training data and, more interestingly, from outside the training data) and get some kind of answer, i.e., some suffix choice.

Why do we pose the problem in these terms? This way of posing the problem leaves a wide range of possible solutions. At the extreme end of one continuum, the end where no significant learning is accomplished, we could build a device that *memorizes* and *fails to generalize*: it jots down on an index card (so to speak), perhaps one for each stem, which combinations of morphosyntactic features were presented to it, along with the correct answer; for these, and only these, the device would be willing to provide a suffix choice; variants on this extremely conservative non-generalizing system could be defined in which a single ('default') suffix is chosen for any stem-feature combination on which the system was not trained.

Such a system is not a good *learner*; it is only a good *memorizer*: it fails to generalize past the data with which it is presented. *Learning*, though, is partly memorization, and partly generalization.

So we turn to a different model of the interaction of input features and (output) suffixes. Rather than assuming that the morphology actively searches the input features and the output suffixes passively await the morphology's decision, we assume rather that *each* suffix actively scans the features that are accessible to it, and makes a quantitative judgment as to whether it is the appropriate suffix to realize those features: the suffix which, in its own quantitative judgment, has the highest activation for a given set of input choices, is the one selected. In the terminology of neural networks, we assume a *winner-takes-all* model of suffix choice: each of the output units has an activation value (by design, a positive value), but only one of the units is considered to have *won*, i.e., will have its morpheme realized—the unit whose activation value is the greatest. This corresponds to a natural machine learning formulation of a classification problem, in which one identifies the appropriate label of a data point by selecting the label-function whose value is greatest, comparing over all label-functions. What are the features that each output unit scans? As we will see, a neural network model allows naturally for two types of organization: one in which the output units have direct access to the input features, and one in which a layer of hidden units intercedes, and the output units have access only to the hidden units, not to the input features *per se*.

There are, however, an unlimited number of models that have such properties, models whose simplicity ranges from the extremely simple to the utterly baroque. In these exploratory experiments, we have looked at a small number of relatively simple models, which we shall turn to in a moment. We will also discuss a proposal developed in [25], which has similar aims to the present

---

<sup>2</sup>We thus abstract away from the two difficult tasks of determining the set of morphosyntactic features, and that of determining the affixes of the language. About the latter, we have much to say; see [17] and other references at <http://linguistica.uchicago.edu>.

paper, though the similarity may only be superficial.

Our primary purpose, however, is one that we hope will not be buried in the details and the implementation, and it is this: we wish to illustrate the notion that if the primary purpose of a linguistic theory is to determine the correct analysis of a finite set of linguistic data (or even to give relative evaluations of competing analyses of the same set of data), then linguistic theory can be viewed as a particular learning model. Two things follow from this: first, the form of the grammar—what it actually looks like—may be influenced by the architecture of the learning theory; and second, some of the basic properties we wish to find in the grammar may be effects (‘artifacts’, some might say) of the learning model itself.

Our ultimate goal (which we do not feel that we have successfully achieved in this paper, but have nonetheless taken some steps toward) is to show how the existence of inflectional patterns is simply another name for generalization in the input-output device that we described above. Neural networks actively generalize, and for reasons we discuss in the next section, we have chosen to explore neural networks as the medium for generalization here, but our purpose is not to argue that neural networks are *the correct way* to model the *stem & feature*  $\rightarrow$  *suffix* relationship, but *a way* to illustrate a style of linguistic analysis which *first* creates a data-analyzer, and only then creates an analysis of a specific set of data.

We can paraphrase our basic problem in this way: Why do some languages choose to use multiple patterns of affixation to express inflectional features, and assign each stem to one of these patterns? Put another way, why don’t languages either act consistently to realize inflectional features the same way for all stems, and why don’t languages allow just *any* combination of affixes to be mixed together to form the inflectional pattern of a stem? Why do these sets of affixes instead cluster together into a small number of often overlapping patterns? Put in these terms, it seems unlikely to us that universal grammar (however one takes that very charged term) comes pre-designed to allow a language to have multiple tableaux of realizations of inflectional paradigms, as in Table 1, up to some fixed limit; rather, the complexity arises because the *data* which the language-learner is confronted with offer (in many cases) two or more suffixes which must be analyzed as bearing exactly the same inflectional features (for example, the *-amos*, *-emos*, and *-imos* of the 1st person plural marker). The cross-linguistic generalization is that when faced with data like this, the language-learner makes a constructive effort to develop an analysis in which the *choices* of the realizations of different parts of the inflectional paradigm are bundled together into patterns that are reused. Our general perspective is that the *cause* of this effect is the essential role played by *dimensionality reduction of the data in all learning*.

At a certain level of abstraction, this style of linguistic analysis is highly reminiscent of the work which Zellig Harris (e.g., in [20]) and others of his generation, proposed: they argued that the goal of a scientific linguistics was the establishment of an explicit method of linguistic analysis, and an analysis of a set of data was only as good as the method which produced it. This position was challenged by Chomsky [5], and largely abandoned in mainstream linguistics in the following decades, for various reasons, not the least being the inadequacy of available learning theories at the time. Times have changed; learning theories are considerably more advanced now; and while they do not typically produce

output (so to speak) in a format that is transparent to the working linguist, we can no longer make the case that learning theories are incapable of dealing with linguistic data: we can only ask *how well* learning theories deal with linguistic data, and whether we have something to learn from how they treat linguistic data. (see [29])

### 3 Neural Networks and Other Learning Models

The analytic work of the linguist—if we were to use the terminology of contemporary machine learning—largely amounts to developing an optimal set of *hidden parameters*. Categories that are not overtly present in the data are, in these terms, *hidden*, and a number of powerful techniques have been developed in order to analyze data using hidden categories; two well-studied techniques are neural networks and hidden Markov models (HMMs). There are others, and there will no doubt be still more in times to come. We will briefly compare and contrast neural networks and HMMs, the first in order to explain what we did, and the second in order to contrast it with an alternative machine learning model that could in principle be applied to the same problem.

Neural networks are composed of large numbers of similar computational units, called *nodes* (or neurons), organized in a particular architecture or pattern of connections—technically, a directed graph—among the nodes. The nodes of the graph are the nodes of the network; the edges of the graph are the connections between the nodes, and the edges are associated with positive or negative numbers, the *weights* of the connections.

Each of the nodes is associated with an *activation level*, a number (in our models, between 0 and 1) which is calculated by summing the activation incoming to the unit, and passing that summed activation through an *activation function*. The activation that is incoming to a given node N is computed in the following way: for each edge E which terminates in N (and hence starts in some other node M), we multiply M’s activation level and E’s weight; and for node N, we sum these products across all edges incoming to N. The activation of node N is related to this sum by an activation function, which could in principle be the identity function, but which is typically a logistic function, such as:

$$\text{logsig}(x) = \frac{e^x}{1 + e^x}$$

This describes the local structure of a neural network. In addition, neural networks typically divide into two sorts, depending on whether their graphs are bipartite (or nearly bipartite) or not. A graph which is bipartite is one for which the nodes can be divided into two disjoint subsets, with the property that all of the edges start in one of the sets and end in the other. Neural networks of this sort are called *feed-forward* networks, and are distinct from other sorts, such as *Hopfield* nets. (see [1], [33])

A simple, two layer feed-forward network, then, is a neural network in which each node can be associated with one of two layers (which we can refer to as Input and Output), and all connections run from Input nodes to Output nodes. When we stop thinking of the network in purely formal terms, we will actually mean something by the terms “input” and “output”; for the moment, they are just names. Since all nodes are functionally equivalent, we can take the Output

layer of one two-layer network and build a new neural network in which it plays the role of Input layer, as in Figure 1.

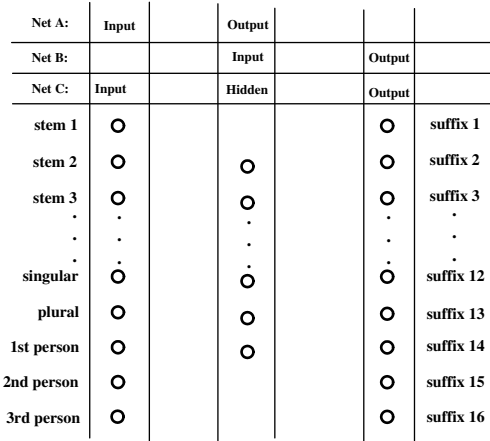


Figure 1: Diagram of neural network

When the architecture of the model has been selected (by the experimenter), we can impose a *meaning* on the nodes in the input and output layers. For nodes in the input layer, we externally decide whether they are turned on (=take an activation of 1.0) or off (=activation of 0.0), and assign an interpretation to each such node. We label one unit as “first person”, for example, and say that when that node is turned on, the network has the symbol “first person” as part of its input. (In this respect, a neural net employs unabashedly symbolic modes of representation.) In a not quite entirely parallel way, we assign meaning to the output nodes, and in the model we explore in this paper, each output node is associated with a particular morph (a morph is a string of phonemes that expresses the phonological aspect of a morpheme). However, in line with much neural network modeling (though not all), we assume a “winner takes all” principle, whereby the selection of an output unit is performed by picking the output unit whose activation is the greatest of all of the nodes on the output layer. Put in another idiom, a winner-takes-all model employs the highly non-linear function  $\text{argmax}(\text{output}_i)$  to select one of the output units based on its activation level, and those activation levels are themselves the result of what is usually a highly non-linear function of the activation levels of units on the preceding layer.

Thus a semantics of a sort is directly associated (by the experimenter) with the nodes on the input and the output layers. No such semantics is associated with units on the hidden layer, in neural networks of the sort illustrated in Figure 1.

Typically one builds neural networks with considerably fewer nodes in the hidden layer than in the input or output layer, because it is the lack of memory resources in the hidden layer that creates the cognitive pressure on the neural network to generalize. Consider a neural network that is forced to learn to associate suffixes with a selected stem and a set of morphosyntactic features, and suppose that there are hundreds of distinct stems modeled in the input

layer. In an architecture with a hidden layer, the output-layer morphemes do not directly perceive the stem-choice nodes; they perceive only an activation pattern over a much smaller set of nodes, those of the hidden layer; and thus to be successful, the neural network must find a way to project onto a *relatively constant pattern* of activation of (some subpart of) the hidden layer units the message that each stem needs to pass on to the suffixes to let them know which suffixes are appropriate to them. All this is a long-winded way of saying that the neural net must find a way of identifying the inflectional classes of the stems with a *pattern of activation* on the hidden layer, because the output nodes do not have direct access to the stem choices (the hidden layer is in the way), and the hidden layer is too impoverished to completely represent precisely which stems have been chosen. From all of this, it follows that neural nets tend to generalize.

In addition, the presence of the hidden layer of nodes can be very reasonably interpreted as the cognitive embodiment of an autonomous linguistic level, one whose patterns of activation correspond to the abstract objects that are related indirectly to both input and output: in the cases that we will look at below, we will find patterns of activation in the hidden units that correspond to inflectional patterns. The take-home message that we observe is that the existence of hidden parameters, such as those associated with the presence of a layer of hidden units in a neural network, are the modeler's embodiment of the abstract grammatical features that the linguist infers from the data.

Neural networks are of interest for several reasons: first, they bear some resemblance, arguably, to the structures used by biological systems for computation; second, they incarnate a relatively simple and comprehensible mathematics; and third, they can relatively easily be endowed with learning abilities. That is, it is possible to calculate connection weights for the edges of a network in a learning context: one presents the neural network with a sequence of training data (the appropriate input nodes are activated, and the desired output nodes are simultaneously activated), and fairly simple, local calculations can be performed to find values for the connection weights that will produce the desired output for the input presented. The real question, of course, is how well the network will *generalize* from the training data to additional data, and we discuss this question in detail below.

Neural networks of the sort we are considering make no pretense to dealing with the sequential character of language—the fact, for example, that suffixes when spoken follow their stem. This is neither good nor bad; it is a fact about the way in which we are abstracting away from the huge task of analyzing all of language to dealing with one small question in morphology.

We would like to very briefly contrast the approach which we have chosen with a short description of an alternative approach which employs an equally standard and familiar technique in machine learning, hidden Markov models (HMMs). As the name suggests, HMMs explicitly learn hidden parameters, and in a fashion not entirely dissimilar from neural networks, do so from training data. Unlike neural networks, HMMs are explicitly probabilistic models. One relatively simple way to describe an HMM is as a finite state automaton (FSA) in which transitions are permitted between any pairs of states, but a probability distribution is assigned to each of the transitions out of a given state; and, in addition, each transition from state M to state N is associated either with (in the simplest case) the emission of a symbol, such as a morpheme or a phoneme)

or (in a more complex case) the “transducing” of one symbol to another. In this latter case, we view the HMM as a formal, finite probabilistic device that finds the “best” way to transduce one string of symbols to a different set of symbols, where “best” means “of highest probability,” and where the probabilities are derived empirically from a training corpus of data.

Thus a path through a particular sequence of states (from the start state to a final state) is associated with a sequence of emitted symbols (e.g., morphemes) or with a sequence of pairs of input/output symbols; more to the point, however, an HMM can easily calculate what is the *most probable* path through its various states that can be associated with the emission of a particular morpheme sequence. To do this, as we have said, it must acquire some set of transition and emission probabilities, and it can do this relatively painlessly on the basis of a training corpus.

One could relatively painlessly construct a hidden Markov model which has a small set of hidden states, in many ways parallel to the neural network that we explore in this paper, and we could test how well the linguistic generalizations were correctly induced by the device. The general point, in our view, is that it behooves us to explore the adequacy with which currently popular models in machine learning (neural networks, HMMs, support vector machines, and so on) can or cannot be applied to linguistic problems in order to determine whether categories required for the adequate description of a language can be induced from the data.

## 4 Previous Work

The present paper looks at some phenomena that were treated by MacWhinney et al [25], who had the explicit goal there of showing that a rule-based approach to the description of language is inferior to a connectionist approach, in which

learning and processing are treated directly in the same computational architecture. As the network learns, its processing abilities develop. There is no separation between learning, structure, and processing. The emphasis is upon maximizing the contribution of the learning algorithm and minimizing the recourse to innate abilities....We are interested in constructing a concrete cue-based alternative to [the generative] approach. (p. 256).

We take a rather different view of both their findings and our own. On the one hand, we believe that MacWhinney et al did not do enough work to show that a connectionist account (as the term would be understood either in 1989 or 2006) was adequate to handle natural language morphology. This is hardly surprising; it would take man-decades at the least to produce a connectionist theory capable of handling Semitic, Bantu, Penutian, and Proto-IndoEuropean morphology, and it might require much more than that. But they did point out something important: that as generative models are almost totally unequipped to provide insight into learning, even a formal model as poorly designed to handle temporally sequential information as connectionism is able to shed important light on learning, when the connectionist conception is coupled to a clear problem of linguistic acquisition.

In the years since MacWhinney et al. [25], the linguistic evidence has continued to grow that morphological categories are genuine aspects of natural language, and that they cannot be dispensed with in favor of semantic or phonetic cues; in some sense, these morphological categories are abstract, but what exactly does that mean? (Aronoff [2] and Spencer [36] are two recent works that discuss the evidence.) If one were building a connectionist model, one might interpret (as we do) the notion of an abstract morphological category as a pattern of activation on a hidden level that plays a consistent causal role in explaining which suffixes are used with which input feature combinations. The work that a connectionist model does is precisely to find the patterns of weights between the hidden units and the non-hidden units (within a relatively constrained space of possibilities, due to the restrictions on the number of hidden units discussed above) which allow for learning and eventually for behavior that adequately reproduces the training data.

There are some flourishes in MacWhinney et al [25] that seem to us irrelevant to the real point at hand. In an effort to emphasize the importance of phonetic and semantic cues for gender assignment in European languages, they suggest strongly (or perhaps state explicitly—the reader is invited to reread the paper) that traditional grammarians were unaware of the importance of phonology, morphology, and semantics in the determination of gender in the Germanic and Romance languages. They point to a paper on child language acquisition (Maratsos and Chalkley [26]) which assumes that gender in German is too arbitrary for a learner to induce useful generalizations; and then they cite a number of papers published during the 1980s which support the claim that phonological, morphological, and semantic features can indeed predict as much as 90 percent of the genders in the first volume of the Duden [8]. But this is not news, nor was it at the time; they do not appear to be aware that the fourth volume of the very same Duden [9] already contains a nuanced presentation of the ways in which phonological, morphological, and semantic features play a major role in influencing or determining a noun’s gender (pp. 137, 146-7); indeed, it would be difficult—perhaps impossible—to find a grammar book, or even a textbook, of a European language which did not illustrate the importance of the role played by morphological and semantic considerations in gender assignment to nouns. This really is not news, and this illustrates the more general point that there truly is a linguistic scholarship that needs to be respected when grammatical issues come under discussion. But the major take-home point that looms over all discussions of the importance of phonological, morphological, and semantic cues in morphology is that while they may be cues, they are not determinative, and that there is a morphology which contains generalizations not immediately reducible to properties in other components. A neural network with a hidden layer is an extremely modest effort to give a location to that (abstract) entity, the inflectional class.

## 5 Our Model of Morphological Classification

Our model is a feed-forward backpropagation neural network with one hidden layer. The training algorithm we use is a Bayesian Regulation backpropagation

algorithm (Matlab’s `trainbr`<sup>3</sup> The hidden and output layers use a log-sigmoid transfer function `logsig`, and the output is run through a competitive algorithm `compet`, in order to determine which suffix should be activated.

The input layer consists of stem nodes and morpho-syntactic information nodes. That is to say, every stem that we consider is given a distinct node in the input layer. Similarly, each morphosyntactic feature value that we use as an input is assigned a node in the input layer. In much the same way, the output layer is made up of suffix nodes, each node representing a phonologically distinct suffix (see Figure 2). Every node of the input layer is connected to every node in the hidden layer (though not every connection is shown in Figure 2); every node of the hidden layer is connected to every node in the output layer (though again, only some of the connections are illustrated).

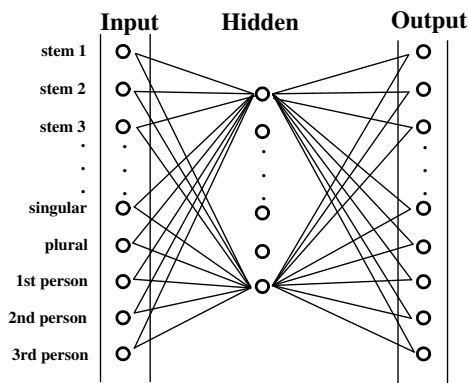


Figure 2: Basic diagram of our model

## 5.1 Spanish Regular Conjugation

In our first experiment with this model, we used Spanish conjugations of regular verbs. Spanish, like other Romance languages, has three principal conjugation classes, named after the suffix used to mark the infinitive: first conjugation (*-ar* verbs), second conjugation (*-er* verbs), and third conjugation (*-ir* verbs). The first conjugation class is the most productive and largest class, but the second and third conjugation classes also make up a significant portion of the Spanish verbal lexicon.

Included in our data set were all six combinations of person and number (1st, 2nd, 3rd person; singular, plural) in the present tense.<sup>4</sup> Additionally,

<sup>3</sup>In Bayesian Regulation, the weight and bias values are determined by Levenburg-Marquardt optimization. For more detailed information see [23], [10]. (source: Matlab help documentation).

<sup>4</sup>Lila Gleitman has pointed out to us that the choice of data in a learning experiment such as this one is likely to have considerable impact on the results, and other hypotheses discussed in the language acquisition literature are likely to play a significant role in the way the present phenomena are handled by young language learners. Newport, for example (see [27], [28]), explores a “less is more” hypothesis, which accounts for an apparent decay in language learning ability which emerges simply out of the maturation of other capacities; Gleitman ([13]

we included the infinitive of each verb. Each token was given in the following format:

**stem, mood, person, number, suffix**

For instance, a token for ‘we sing’ would look like this:

**cant, ind, 1p, pl, amos**

The stem for ‘to sing’ is *cant-* (infinitive *cantar*), the mood is indicative (**ind**), and it is conjugated for the first person plural (**1p, pl**). The appropriate suffix in this situation is *-amos*, giving a complete inflected form of *cantamos*, ‘we sing’.

In the example above, four input nodes would be activated (those corresponding to **cant**, **ind**, **1p**, and **pl**), and the desired result would be that the output neuron associated with **amos** would be the node whose activation is the greatest among all the nodes in the output layer. See Figure 3 for a graphical representation of this example token.

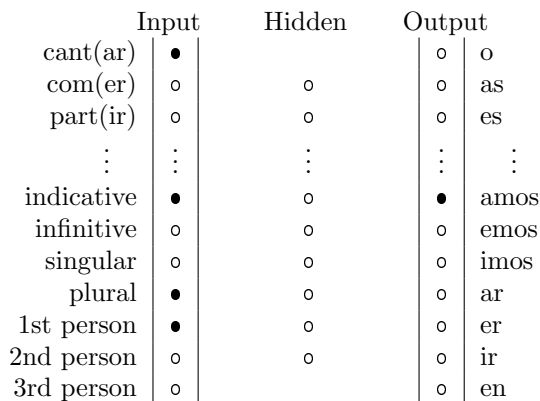


Figure 3: Diagram of the token *cantamos*

In our experiment, there were 37 input nodes: 30 were associated with stems (10 of each conjugation class), and 7 were associated with morpho-syntactic items (**1p**, **2p**, **3p**, **sg**, **pl**, **ind**, **inf**). There were 16 output nodes, each corresponding to a (phonologically) distinct suffix, or *morph*. The distribution of these suffixes can be seen below in Table 2.

For inputs and targets, the network was given 15 verb stems (5 of each class), each verb stem fully conjugated (resulting in 7 conjugated forms for each verb stem). We gave the network 15 additional verb stems (also 5 of each class), but these verb stems were only conjugated in the infinitive in the training data. The network was trained using the Matlab `train` command until the performance gradient fell below  $1 \times 10^{-10}$ , which usually took approximately 50 epochs.

After this training period, a simulation was run with the network, using the input data for all 30 verbs, fully conjugated. When prompted for the suffixes, the network was able to reproduce all of the training data with 100% accuracy.

---

and elsewhere) accounts for differences between categories of words with varying degrees of concreteness in terms of word-to-world pairings. If we focused on the inability of formal models to learn, then it would be especially important for us to consider a more careful organization of the data presented to it. Since our results focus on the success the formal model had, we are less concerned, especially at this early stage; but the point is well taken.

	-ar (1st)	-er (2nd)	-ir (3rd)
1.sg	-o	-o	-o
2.sg	-as	-es	-es
3.sg	-a	-e	-e
1.pl	-amos	-emos	-imos
2.pl	-áis	-éis	-ís
3.pl	-an	-en	-en

Table 2: Spanish verb conjugations for the present tense

When the network was asked to generalize to the 15 verb stems whose infinitives it was trained on, it had a 94% success rate (5 tokens incorrect / 90 new tokens).

This success rate depended on the number of nodes in the hidden layer. The network with a 94% success rate (6% error rate) had ten hidden nodes. As can be seen in Figure 4, if the number of hidden nodes is too small, then the network will not perform optimally.

For an explanation of what the hidden nodes are actually doing in the network, see the discussion in Section 5.

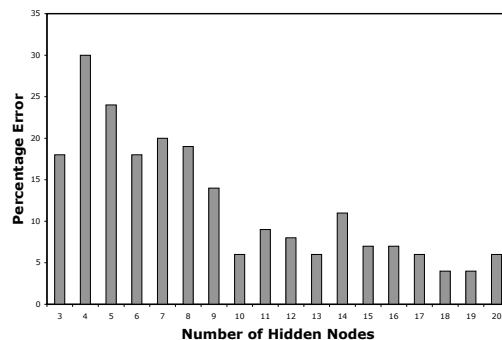


Figure 4: Spanish verb conjugation: number of hidden nodes vs. performance  
(The results given are the average of five trials for each number of hidden nodes.  
Each trial was run on a separate network with different random initial weight values.)

## 5.2 German Definite Articles

Unlike the English equivalent *the*, the German definite article declines based on case, number, and gender. In the experiment with Spanish, our goal was to have the network be able to distinguish between different verb conjugation classes, learning the abstract category of word class from the data itself. In much the same way, we wanted this type of network to be able to learn the abstract category of gender, generalizing only from the noun stems and the sets of forms

of the definite article that each stem takes. Thus, gender in this experiment corresponds precisely to conjugation class in the previous experiment.

Instead of training the network to give the suffix of the noun, in this experiment we trained the network to give the corresponding definite article. As in the previous experiment, we gave the network 30 stems (10 of each gender), half of them with the definite article fully declined, the other half with just the nominative singular article given. There is no special status for nominative singular; we could have easily chosen the accusative singular, as both forms will distinguish the three genders. Each token was given in the following format:

**stem, case, number, definite-article**

For instance, a token for ‘the men’ (nominative) would look like this:

**Mann, nom, pl, die**

The stem for ‘man’ is *Mann*, the case is nominative (**nom**), and it is in the plural (**pl**). The corresponding definite article in this context is *die*. Like in the Spanish experiment, each stem and morpho-syntactic item has a corresponding node in the input layer, and each of the six definite articles has a corresponding node in the output layer. As can be seen in Table 3, the definite article can be realized as one of the following: *der*, *den*, *dem*, *des*, *die*, or *das*. The definite article for all plurals declines the same way, regardless of gender.

	masc.	fem.	neuter	plural
nom	der	die	das	die
acc	den	die	das	die
dat	dem	der	dem	den
gen	des	der	des	der

Table 3: Declension of the German definite article

The training of this network was completed using the same function and settings as in the Spanish experiment.<sup>5</sup> Just as in the Spanish example, the network was able to reproduce all of the training data completely, and was able to generalize rather well to the nouns that had not been declined, with an accuracy of 90%. Again, the ability of the network to generalize is related to the number of hidden nodes. Varying the number of nodes will alter the performance of the network (see Figure 5).

### 5.3 German Noun Suffixes and Umlaut

The story of the German case system does not end with article declension. In addition to case, number, and gender being marked on the definite article, this morpho-syntactic information is also marked on the noun itself, in the form of suffixes and systematic vowel changes (umlaut). The noun suffixes used depend on which declension class and subclass the noun belongs to (see Table 4 for further details). The noun suffixes also partially depend on gender, inasmuch as feminine nouns do not decline in the singular.

The nouns chosen were the first 30 nouns of a Swadesh list<sup>6</sup>. We only chose those nouns that had a plural form (e.g. *Blut* ‘blood’ does not have a plural

<sup>5</sup>Sometimes, the network would not converge, in which case we would run the network for 100 epochs and then test it, regardless of its performance gradient.

<sup>6</sup>A Swadesh list, first devised by Morris Swadesh [37], is a list of the most commonly

	I		II		III		IV	
	Masc & Neuter	Fem	Masc & Neuter	Fem	Masc & Neuter	Masc & Neu	Feminine	
	w/o " with "	with "	w/o " with "	with "	w/o " with "	with -e-	w/o -e- with -e-	
nom.sg	-	-	-	-	-	-	-	-
acc.sg	-	-	-	-	-	-en	-	-
dat.sg	-	-	-	-	-	-en	-	-
gen.sg	-s	-	-es	-	-(e)s	-en	-	-
nom.pl	-	..	-e	..e	-er	-er	-n	-en
acc.pl	-	..	-e	..e	-er	-en	-n	-en
dat.pl	-	..n	-en	..en	-ern	-en	-n	-en
gen.pl	-	..	-e	..e	-er	-er	-n	-en

**Examples**

	I		II		III		IV	
	Masc & Neuter w/o "	Fem with "	Masc & Neuter w/o "	Fem with "	Masc & Neuter w/o "	Fem with "	Masc & Neu with -e-	Feminine w/o -e- with -e-
nom.sg	Knochen	Vater	Hund	Baum	Kind	Mann	Mensch	Schlange
acc.sg	Knochen	Vater	Hund	Baum	Kind	Mann	Menschen	Schlange
dat.sg	Knochen	Vater	Hund	Baum	Kind	Mann	Menschen	Schlange
gen.sg	Knochens	Vaters	Hundes	Baumes	Kindes	Mannes	Menschen	Schlange
nom.pl	Knochen	Väter	Hunde	Bäume	Kinder	Männer	Menschen	Schlangen
acc.pl	Knochen	Väter	Hunde	Bäume	Kinder	Männer	Menschen	Schlangen
dat.p	Knochen	Vätern	Hunden	Bäumen	Kindern	Männern	Menschen	Schlangen
gen.pl	Knochen	Väter	Hunde	Bäume	Kinder	Männer	Menschen	Schlangen

Table 4: Declension of German noun classes.

The classes shown are only the ones in the training data. There are more classes and subclasses in the complete German grammar.

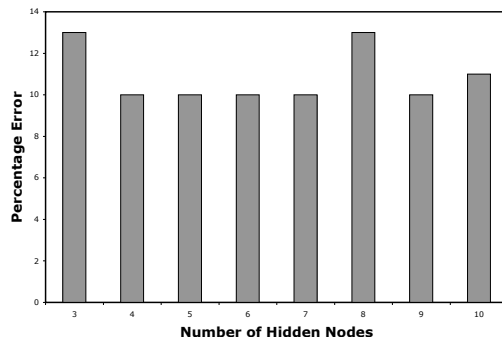


Figure 5: German article declension: number of hidden nodes vs. performance (The results given are the average of five trials for each number of hidden nodes.)

form). The first 17 nouns were fully declined, and the remaining 13 nouns were used to verify the network’s ability to generalize.

A token was represented in the following way:

**stem, case, number, suffix, vowel-change**

For instance, a token for ‘forests’ (nominative) would look like this:

**Wald, nom, pl, er, x**

*Wald* is the stem form (and therefore also the nominative singular form) of ‘forest’. The suffix is *-er*, and the main vowel of the stem undergoes umlaut, represented in German orthography by a diacritic mark “ placed over the vowel. Thus, the nominative plural of ‘forest’ is *Wälder*. If there had been no umlaut, then there would have been no ‘x’ in the training data.

The network performed quite well in our simulations, correctly repeating 100% of the training data, and generalizing to the new data with 97% accuracy. See Figure 5 for details concerning the relationship between the number of hidden nodes and performance of the network.

## 6 Discussion

How can the network generalize as well as they do? A number of factors contribute to the answer, but the primary reason is the hidden layer of nodes.

As was explained in Section 2, the hidden layer allows the network to better organize and categorize stems into inflectional classes. When the network is trained to output *one* particular form (the infinitive, in the example with Spanish), the other forms of the inflectional class are also favored, and therefore the network can predict with very high accuracy all of the other inflected forms. Even so, the exact system by which this happens is not entirely known.

When we began these experiments, we naively thought that in an ideal system, one hidden node would correspond to *-ar* verbs, another would correspond

---

used words in a language. Swadesh used it to determine how close a language might be to any other language. We chose this method because it gives a higher priority to word frequency than word class or regularity. Example Swadesh lists can be found at [en.wiktionary.com/wiki/Swadesh\\_List](http://en.wiktionary.com/wiki/Swadesh_List).

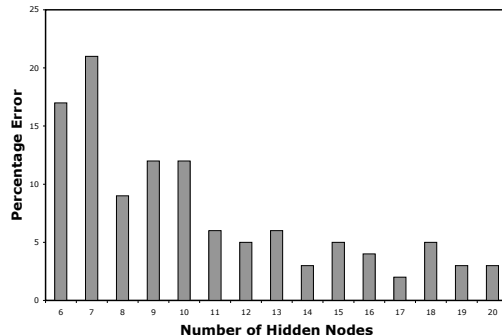


Figure 6: German noun declension: number of hidden nodes vs. performance (The results given are the average of five trials for each number of hidden nodes.)

to *-er* verbs, and a third to *-ir* verbs. This is not how things play out in actual neural networks. Stems of the same inflectional class will connect to the hidden nodes in a similar way, but there are no dedicated nodes for inflectional class. Instead, inflectional categorization appears to be a function that is distributed among a number of the hidden nodes. We will first take a look at the way in which this happens, and then discuss its significance.

## 6.1 The workings of the hidden layer

Table 5 helps explain how the hidden layer actually works. Each entry in the table is given in the format `input-node:weight`. For each of the 9 hidden nodes, this table gives the 13 weights with the highest absolute value. In other words, every hidden node has a weight that connects it to each of the 37 input nodes, and this table shows some of those weight values.

Entries in italics contain negative weights, and entries in boldface are from morpho-syntactic information nodes (i.e. `1p`, `2p`, `3p`, `sg`, `pl`, `ind`, `inf`). The superscript numbers indicate to which conjugation class the verb belongs. For instance, in the very first entry *c<sup>1</sup>:-5.7*, the superscript 1 stands for the first conjugation class (*-ar* verbs), and the *c* stands for the stem of the verb *cantar* ‘sing’. The weight between this input node (`cant`) and the first hidden node is  $-5.7$ . As another example, the weight between the second hidden node and the third person input node (`3p`) is 6.4, and this is represented by the entry **`3p:6.4`**. Therefore, if the `3p` input node is activated (which is a value of 1), then the second hidden node would be activated to a value of  $1 \times 6.4 = 6.4$ , the product of the activation of the input node and the weight.<sup>7</sup>

Note how verb stems of the same conjugation class tend to stick together. This is because they are trained on the same data, and therefore they each connect to the same hidden nodes in the same way. There are 10 stems from each conjugation class in this network, but often we find only 5 of them together. The reason is that these are the 5 stems that the network was trained on (i.e.,

<sup>7</sup>This value would then be added to the activations from all the other input nodes and added to the bias, then this sum would be computed with the log-sigmoid function to determine the end activation value of the hidden node.

node	weights to corresponding input nodes											
1	$c^1:-5.7$	$p^1:-5.7$	$p^1:-5.7$	$e^1:-5.7$	$m^1:-5.7$	<b>sg:4.9</b>	<b>inf:-4.6</b>	<b>3p:-4.3</b>	<b>2p:4.1</b>	$d^2:3.3$	$le^2:3.3$	$co^2:3.3$
2	<b>2p:-7.6</b>	<b>3p:6.4</b>	<b>1p:2.0</b>	<b>pl:1.6</b>	$db^3:1.4$	$pt^3:1.4$	$pb^3:1.4$	$c^3:1.4$	$dd^3:1.4$	$ad^3:1.3$	$ac^3:1.3$	$b^3:1.3$
3	<b>pl:-6.3</b>	<b>sg:6.3</b>	$cm^2:4.8$	$cm^2:4.8$	$le^2:4.8$	$cr^2:4.8$	$co^2:4.8$	<b>3p:-4.1</b>	$as^3:-3.8$	$ap^3:-3.8$	$ad^3:-3.8$	$ac^3:-3.8$
4	<b>sg:6.3</b>	<b>pl:-5.5</b>	<b>2p:4.5</b>	$co^2:-4.2$	$cm^2:-4.2$	$le^2:-4.2$	$d^2:-4.2$	$cr^2:-4.2$	<b>3p:-4.1</b>	$p^1:3.7$	$c^1:3.7$	$m^1:3.7$
5	<b>1p:-9.4</b>	<b>2p:5.4</b>	$c^1:4.9$	$p^1:4.9$	$b^1:4.9$	$m^1:4.9$	$e^1:4.9$	<b>sg:-4.1</b>	<b>pl:2.3</b>	<b>3p:2.1</b>	$pt^3:-2.0$	$pb^3:-2.0$
6	$pt^3:5.7$	$pb^3:5.7$	$c^3:5.7$	$dd^3:5.7$	$db^3:5.7$	$s^2:-4.5$	$bb^2:-4.5$	$br^2:-4.5$	$a^2:-4.5$	$v^2:-4.5$	<b>sg:-4.4</b>	$le^2:-4.2$
7	<b>1p:-7.4</b>	$c^1:-6.2$	$b^1:-6.2$	$m^1:-6.2$	$e^1:-6.2$	$p^1:-6.2$	<b>3p:4.7</b>	$g^1:-4.1$	$r^1:-4.1$	$i^1:-4.1$	$hl^1:-4.1$	$hb^1:-4.1$
8	<b>sg:-6.7</b>	<b>pl:6.1</b>	$c^3:-4.1$	$dd^3:-4.1$	$db^3:-4.1$	$pt^3:-4.1$	$pb^3:-4.1$	$as^3:-2.0$	$b^3:-2.0$	$ac^3:-2.0$	$ad^3:-2.0$	$ap^3:-2.0$
9	<b>1p:7.0</b>	<b>3p:-5.4</b>	<b>2p:-5.3</b>	<b>inf:4.9</b>	<b>pl:-4.8</b>	<b>ind:-3.7</b>	$c^1:2.4$	$b^1:2.4$	$m^1:2.4$	$e^1:2.4$	$p^1:2.4$	$cm^2:-2.1$

Table 5: Weights between input nodes and hidden nodes in the Spanish experiment

the ones that were given all the conjugated forms). For the first hidden node, the 5 verb stems at the beginning (each with a weight value of  $-5.7$ ) are the first conjugation stems on which the network was trained; the other 5 first conjugation stems connect to the hidden layer in a slightly different way. Sometimes they all connect in the same way, as can be seen with the third conjugation stems and the second hidden node (all with a weight value of 1.4 or 1.3).

Note also that no hidden node has a single dedicated task; the hidden layer distributes the categorization among all nine hidden nodes. For instance, the ninth hidden node tends to be used for the first person (**1p:7.0**), and it also is used for infinitives (**inf:4.9**) and first conjugation verbs ( $c^1:2.4$ ,  $b^1:2.4$ , etc.).

## 6.2 The significance for learning of using vectorial representation for symbols

What we have referred to as a “distributed” representation (of inflectional classes, in the cases considered here) in the discussion so far is better understood as a vectorial representation of information. As is well-known (and has been known as long as there have been neural networks), the state of a neural network is best thought of as a location in an  $n$ -dimensional space, where  $n$  is the number of nodes in a particular layer, and the location is normally represented as a vector in  $n$ -space. There are many conceptual advantages to this way of thinking about the activation states of networks. Not the least of these is that it gives one no reason to expect that activation of a *single* unit (corresponding to a vector of the form  $(0,0,\dots,1,\dots,0,0)$ ) should be of any particular importance—which is to say, no reason to expect that a single node should correspond to some grammatical property like inflectional class; functions are associated with whole patterns of activation, which is to say, conceptually associated with a point in a vector space.

## 6.3 The significance for grammar of worrying about learning

One conclusion that we draw from this study is that there is a danger inherent in assuming that any particular analytic method, such as traditional linguistic analysis, rests on a solid enough bedrock of certainty that we can trust it to tell us specifically what the adult state of linguistic knowledge is with certainty. Early on, we expected that the traditional tableau of a verbal paradigm would be directly represented in units of the connectionist network, but we were wrong. The conclusion we draw from this is that learning theory (in all its guises, both computational and developmental) is best thought of as being in a position to negotiate with linguistic theory (and vision theory, and so on); neither set the agenda inflexibly. Some styles of grammatical description lend themselves poorly, if at all, to being integrated into theories of learning, and such descriptions may lead grammarians to conclude that assumptions that they make are innately known, *because* they are apparently unlearnable. We view the efforts described in this paper as valid steps towards a working compromise between contemporary learning theory and grammatical theory.

## 6.4 The vanishing of the inflectional tableau from the analysis

From a linguistic point of view, there is one abstraction that looms large in traditional descriptions of inflectional morphology that does not appear overtly in the present connectionist model, and that is the multidimensional array or matrix illustrated in Table 1 and referred to by Aronoff in the quotation in Section 2 above. That array is spanned by morphosyntactic features, the inputs to the hidden units in this model, and the cells of that array are filled by morphs, which are directly associated with the output units of our present model. Because of the presence of the hidden units, whose activation patterns simulate inflectional classes, what we have conceptually in the pairing of the hidden units and the output units is a vector space of the same dimensionality as the number of hidden units, and that space is cut up into  $P$  sub-volumes, each of which causes one of the  $P$  different output morphs to be the winner. That geometry is what is responsible for morphs often being associated with several cells in the traditional morphosyntactic feature array, the phenomenon known as syncretism, as noted above. In the present model, syncretism arises without anything specific needed to be said about it (see [16] for further discussion of this).

## 6.5 Could this be all wrong?

A reviewer for this journal raised the following question: what if the model presented here is *just wrong*? — and indeed, this is a question that we have not addressed thus far. In our view, it is a near certainty that the model is *wrong*, but it is very unlikely that it is *just* wrong. There are various ways in which it could fail: it could fail to “scale up” (fail in its treatment of much larger sets of data); it could fail to “scale over” (fail to work at all to data from other languages which seemed to be parallel to the data considered here); it could fail to merge well with other computational modeling aspects of language acquisition, such as the modeling of segmentation and classification (see references in Section 1 above); it could be overtaken by a different model, either connectionist or not, which handles a wider range of data, or the same data but in a more effective or efficient way. Any of these outcomes would be solid indications that the model was wrong. The principal way that the model could turn out to be *just* wrong in the long run would be if the correct model of inflectional classes did not employ distributed (vector-based) mathematical models, and if it postulated explicit positions built a priori into the model for a certain number of distinct inflectional classes. Time will tell.

## 7 Conclusion

Our goal in this note has been to explore the way in which well known learning algorithms, such as those associated with feed-forward neural networks, can shed light on questions of language acquisition. At the same time, we have offered an interpretation of the networks’ behavior that is consistent with viewing them as embodiments of traditional grammatical descriptions.

## References

- [1] Daniel J. Amit. *Modeling Brain Function*. Cambridge University Press, Cambridge, UK, 1989.
- [2] Mark Aronoff. *Morphology By Itself*. MIT Press, Cambridge MA, 1994.
- [3] Michael R. Brent. An efficient, probabilistically sound algorithm for segmentation and word discovery. *Mach. Learn.*, 34(1-3):71–105, 1999.
- [4] Andrew Carstairs. *Allomorphy in Inflection*. Croom Helm, London, 1987.
- [5] Noam Chomsky. *Syntactic Structures*. Mouton, The Hague, 1957.
- [6] Mathias Creutz. Unsupervised segmentation of words using prior distributions of morph length and frequency. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 280–287, Morristown, NJ, USA, 2003. Association for Computational Linguistics.
- [7] Carl de Marcken. *Unsupervised Language Acquisition*. PhD thesis, MIT, Cambridge, MA, 1996.
- [8] Paul Grebe et al. *Der Grosse Duden, Volume 1*. Bibliographisches Institut, Mannheim, 1966.
- [9] Paul Grebe et al. *Der Grosse Duden, Volume 4*. Bibliographisches Institut, Mannheim, 1966.
- [10] D. Foresee and M. Hagan. Gauss-newton approximation to bayesian learning. *Proceedings of the International Joint Conference on Neural Networks*, pages 1930–1935, 1997.
- [11] Michael Gasser. Acquiring receptive morphology: a connectionist model. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 279–286, Morristown, NJ, USA, 1994. Association for Computational Linguistics.
- [12] Eric Gaussier. Unsupervised learning of derivational morphology from inflectional lexicons. *Proceedings of the Workshop on Unsupervised Learning in Natural Language Processing*, pages 24–30, 1999.
- [13] Lila Gleitman. Verbs of a feather flock together ii. In Bruce Nevin, editor, *The Legacy of Zellig Harris: Language and information into the 21st century, Volume 1*, pages 206–229. John Benjamins, Philadelphia, 2002.
- [14] John Goldsmith. Local modeling in phonology. In S. Davis, editor, *Connectionism: theory and practice*, pages 229–246. Oxford University Press, Oxford, 1992.
- [15] John Goldsmith. A dynamic computational model of accent systems. In Jennifer Cole and Charles Kisseberth, editors, *Perspectives in Phonology*, pages 1–28. CSLI, Stanford, 1994.

- [16] John Goldsmith. Grammar within a neural net. In Sue Lima Pamela Downing and Gregory Iverson, editors, *The Reality of Linguistic Rules*, pages 95–113. John Benjamins, Philadelphia, 1995.
- [17] John Goldsmith. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198, 2001.
- [18] John Goldsmith. An algorithm for the unsupervised learning of morphology. *Natural Language Engineering*, 12:1–19, 2006.
- [19] John Goldsmith and Gary Larson. Local modeling and syllabification. In K. Deaton K. M. Ziolkowski and M. Noske, editors, *Papers from the 26th regional meeting of the Chicago Linguistic Society, Volume 2*, pages 129–142. Chicago Linguistic Society, Chicago, 1990.
- [20] Zellig Harris. *Methods in Structural Linguistics*. University of Chicago Press, Chicago, 1951.
- [21] Martin Haspelmath. *Understanding Morphology*. Understanding Language Series. Arnold, London, 2002.
- [22] Simon D. Levy and Jordan B. Pollack. Escape the building-block/rule dichotomy: a case study. In *AAAI Spring Symposium on Computational Synthesis*, 2003.
- [23] David J.C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4(3):415–447, 1992.
- [24] Brian MacWhinney. Connectionism and language learning. In *Usage-Based Models of Language*, pages 121–149. CSLI Publications, Stanford, CA, 2000.
- [25] Brian MacWhinney, Jared Leinbach, Roman Taraban, and Janet McDonald. Language learning: Cues or rules? *Journal of Memory and Language*, 28:255–287, 1989.
- [26] Michael Maratsos and Mary Anne Chalkley. The internal language of children’s syntax: the ontogenesis and representation of syntactic categories. In K. Nelson, editor, *Children’s Language Volume 2*, pages 127–214. Gardner Press, New York, 1980.
- [27] Elissa L. Newport. Constraints on learning and their role in language acquisition: studies of the acquisition of american sign language. *Language Sciences*, 10(1):147–172, 1988.
- [28] Elissa L. Newport. Maturational constraints on language learning. *Cognitive Science*, 14:11–28, 1990.
- [29] Fernando Pereira. Formal grammar and information theory: Together again? *Philosophical Transactions of the Royal Society*, 358:1239–1253, 2000.
- [30] Steven Pinker. *Words and Rules*. Basic Books, New York, 1999.

- [31] Vito Pirrelli, B. Calderone, I. Herreros, and M. Virgilio. Non-locality all the way through. emergent global constraints in the italian morphological lexicon. *Association for Computations Linguistics Group on Computational Phonology (SIGPHON)*, Barcelona, July 2004.
- [32] Jordan B. Pollack. Recursive distributed representations. *Artificial Intelligence*, 26:77–105, 1990.
- [33] David E. Rumelhart and James L. McClelland. *Parallel Distributed Processing, Volume 2*. MIT Press, Cambridge, MA, 1986.
- [34] Paul Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist networks. *Artificial Intelligence*, 46:159–216, 1990.
- [35] Paul Smolensky. Tensor product representations: Formal foundations. In Paul Smolensky and Galdine Legendre, editors, *The Harmonic Mind: From Neural Computation to Optimality-Theoretic Grammar*, pages 259–334. MIT Press, Cambridge MA, 2006.
- [36] Andrew Spencer. *Morphological Theory*. Basil Blackwell, Oxford, 1991.
- [37] Morris Swadesh. Salish internal relationships. *International Journal of American Linguistics*, 12:157–167, 1950.