*Cognition*, **48**, 1993, pp. 71-99.

# Learning and development in neural networks: The importance of starting small

Jeffrey L. Elman
Departments of Cognitive Science and Linguistics
University of California, San Diego

## INTRODUCTION

Humans differ from other species along many dimensions, but two are particularly noteworthy. Humans display an exceptional capacity to learn; and humans are remarkable for the unusually long time it takes to reach maturity. The adaptive advantage of learning is clear, and it may be argued that, through culture, learning has created the basis for a non-genetically based transmission of behaviors which may accelerate the evolution of our species. The adaptive consequences of lengthy development, on the other hand, seem to be purely negative. Infancy and childhood are times of great vulnerability for the young, and severely restrict the range of activities of the adults who must care for and protect their young. It is difficult to understand why evolutionary pressures would not therefore prune a long period of immaturity from our species.

It is important to remember, however, that evolution selects for the fitness of whole individuals, not for the value of isolated traits such as an enhanced ability to learn or the duration of infancy. The adaptive success of individuals is determined by the joint interaction of all their traits. So it may be that to understand the perseverance of one trait with apparently negative consequences (such as a lengthy period of immaturity), we need to consider possible interactions with other traits (such as ability to learn). One reason to suspect such an interaction, *a priori*, is that in humans the greatest learning occurs at precisely the point in time—childhood—when they are undergoing the most dramatic maturational changes.

In fact, I would like to propose that in humans, learning and development interact in an important and non-obvious way. Maturational changes may provide the enabling conditions which allow learning to be most effective. My argument will be indirect, based on findings that are obtained with artificial neural network models of learning. There are circumstances in which these models work best (and in some cases, only work at all) when they are forced to “start small” and to undergo a developmental change which resembles the increase in working memory which also occurs over time in children. This effect occurs because the learning mechanism in such systems has specific shortcomings which are neatly compensated for when the initial learning phase takes places with restricted capacity. In this light, a long period of development plays a positive role in the acquisition of a behavior.

This paper is divided into two major sections. I begin by reporting the results of the simulations with artificial neural networks. The goal of these simulations was to train networks to process complex sentences in order to test their ability to learn and to represent part/whole relationships and embedded clauses. The networks were only able to learn the task when they were handicapped by being forced to begin with severe memory limitations. These have the effect of restricting the range of data they were exposed to during initial learning. Thus, the "importance of starting small."

However, this result contrasts with other findings in the connectionist literature. It is known, for instance, that there are problems which can best be learned when the entire data set is made available to a network (Harris, 1991). If a network is given only a subset of the data, it often fails to learn the correct generalization and remains stuck in a local error minimum. This result is thus just the opposite of the starting small finding; instead, it seems sometimes to be necessary to "start big."

This apparent paradox leads to the second part of the paper. Here, I attempt to understand the deeper principles which underlie learning in the general class of connectionist systems which rely on error-driven gradient descent techniques. These principles explain why it is that sometimes it is necessary to start small, and at other times, start big. More basically, we see that these principles of learning interact with characteristics of human development in a beneficial manner.

## PART I: The importance of starting small

One of the most studied domains in which humans learn is language. It is also one of the most theoretically problematic domains for understanding learning, in part because of what has been called the "projection problem." The problem is just that, if the task of the language learner is to figure out the underlying regularities—that is, the grammar—which are responsible for the language he or she hears, then the data which are available to the learner may not be sufficient to uniquely determine the correct grammar.[1]

This problem of the apparent insufficiency of the data has been discussed in many contexts (e.g., Baker, 1979; Bowerman, 1987; Pinker, 1989; Wexler & Cullicover; 1980) but one of the simplest demonstrations comes from Gold's (1967) work. Gold shows that if a language learner is presented with positive-only data, only *regular* languages can be learned (regular languages are languages which can be generated by finite state automata). The rub is that, on the one hand, natural languages appear to belong to a more powerful class than this (Chomsky, 1957); and on the other, there is no good evidence that children receive or use negative data during learning (Brown & Hanlon, 1970; see also Pinker, 1989, for discussion of competing claims).

Gold advances several suggestions in order to account for the fact that, despite his findings, children *do* learn language. Although children do not appear to receive explicit negative evidence,

[1] I say 'may' because much hinges on exactly what one believes the nature of the input to be: bare sentence strings? strings accompanied by semantic interpretations? strings accompanied by information about the environment in which they were uttered?

they may receive indirect negative evidence. Or possibly, some of what children know is innate; thus they need not infer the grammar solely on the basis of positive data.[1]

Almost certainly both of the possibilities outlined by Gold are true to some extent. That is, the child is not an unconstrained learning mechanism in the sense of being able to learn any and all possible languages. Rather, innate predispositions narrow the range of what can be learned. Of course, it is very much an open (and controversial) question exactly what form that innate knowledge takes. A number of investigators have also proposed that although *direct* negative evidence may not be available, there are subtler forms of negative evidence. For example, the non-occurrence of an expected form constitutes an *indirect* sort of negative evidence. Just how far this sort of evidence can be used has been challenged (Pinker, 1989). Thus, although innateness and indirect evidence plausibly participate in the solution of the learnability problem, their contribution is not known and remains controversial.

I would like to suggest that there may be a third factor in helping account for the apparent ability of learners to go beyond the data. This factor hinges on the simple fact that first language learners (children) are themselves undergoing significant developmental changes during precisely the same time that they are learning language. Indeed, language learning after these developmental changes have been completed seems to be far less successful. This is often attributed to the passing of a “critical period” for language learning. But this is no more than a restatement of facts. What I would like to consider here is the question of what it is about the so-called critical period that might facilitate learning language.

Interestingly, much of the work in learnability theory neglects the fact that learning and development co-occur. An especially relevant exception, as we shall see, is Newport’s (1988, 1990) “less is more” proposal, which is very consistent with the results obtained here. The typical assumption is that both the learning device and training input are static. One might wonder what the consequences are of having either the learning device (network or child) or the input data not be constant during learning. Plunkett and Marchman (1990) have shown that while the basic influences of type/token frequency and phonological predictability are similar to the condition of non-incremental learning, better overall learning is achieved when the training corpus for a connectionist model is allowed to slowly grow in size. We might also ask what the consequences are when the learning mechanism itself changing. Allowing networks to dynamically reconfigure our acquire additional nodes has been shown to facilitate learning (Ash, 1989; Fahlman & Lebiere, 1990; Shultz & Schmidt, 1991).

In this section, I report the effect of staged input on learning in a connectionist model. The network fails to learn the task when the entire data set is presented all at once, but succeeds when the data are presented incrementally. I then show how similar effects can be obtained by the more realistic assumption that the input is held constant, but the learning mechanism itself undergoes developmental changes. Finally, I examine the network to see what the mechanism is which allows

[1] Gold mentions a third possibility, which is that if the text is ordered, then positive-only presentation is sufficient to learn even the most complex set of languages he considers. The details of this proposal are not well-developed however.

this to happen and suggest what conditions are necessary for incremental learning to be useful.

## Simulations

This work was originally motivated by an interest in studying ways in which connectionist networks might use distributed representations to encode complex, hierarchically organized information. By this I mean just the sort of relationships which typically occur in language. For example, in the sentence *The girls who the teacher has picked for the play which will be produced next month practice every afternoon*, there are several events which are described. Some are backgrounded or subordinate to the main event. This subordination has grammatical consequences. Thus, the main verb (*practice*) is in the plural because it agrees with *the girls* (not *the teacher*, nor *the play*). And although *picked* is a transitive verb which often takes a direct object following it, no noun appears after the verb because the direct object (*the girls*) has already been mentioned.

These sorts of facts (and specifically, the recursive nature of embedded relative clauses) have led many linguists to conclude that natural language can not be modeled by a finite state grammar (Chomsky, 1957), and that statistical inference is not a viable mechanism for learning language (Miller & Chomsky, 1963). Connectionist networks, however, are not finite state machines. So it is reasonable to wonder ask connectionist networks (which nonetheless rely heavily, though not exclusively, on statistical inference) possess the requisite computational properties for modeling those aspects of natural language which are beyond the processing capacity of finite state automata.

To address this question, I constructed a semi-realistic artificial language which had some of the crucial properties which are problematic for finite state automata and statistical learning, and then attempted to train a neural network to process sentences from this language. The particular network employed was a simple recurrent network (Elman, 1990, 1991; Jordan, 1986; Servan-Schreiber, Cleeremans, & McClelland, 1988). The salient property of this architecture is that the internal states (in this case, hidden unit activation patterns) are fed back at every time step to provide an additional input. This recurrence gives the network dynamical properties which make it possible for the network to process sequential inputs. Exactly *how* the sequential information is represented in the internal states is not determined in advance, since at the outset of training connection weights (and activation states) are random. Instead, the network must discover the underlying temporal structure of the task and learn to encode that structure internally. The network architecture that was used is shown in Figure 1.

*— Insert Figure 1 about here —*

The input corpus consisted of sentences which were generated by a grammar that had certain critical properties: (a) subject nouns and their verbs agreed for number; (b) verbs differed with regard to argument expectations; some verbs required direct objects, others optionally permitted objects, and others precluded direct objects; and (c) sentences could contain multiple embeddings in the form of relative clauses (in which the head could be either the subject or object of the subordinate clause). The existence of these relative clauses considerably complicated the set of agreement and verb argument facts. Sample sentences are given in Table 1.

An important aspect of training was that words were represented in a manner which did not convey information about grammatical category, meaning, or number. Each word was encoded as a vector of 0's in which a single bit was randomly set to 1. Thus, not only the grammatical structure but also the nature of the different inputs was obscure to the network.

The network was trained to take one word at a time and predict what the next word would be. Because the predictions depend on the grammatical structure (which may involve multiple embeddings), the prediction tasks forces the network to develop internal representations which encode the relevant grammatical information. (See Elman, 1991, for details of this language.)

boys who chase dogs see girls.
girl who boys who feed cats walk.
cats chase dogs.
mary feeds john.
dogs see boys who cats who mary feeds chase.

Table 1

The results of the first trials were quite disappointing. The network failed to master the task, even for the training data. Performance was not uniformly bad. Indeed, in some sentences, the network would correctly coordinate the number of the main clause subject, mentioned early in a sentence, with the number of the main clause verb, mentioned after many embedded relative clauses. But it would then fail to get the agreement correct on some of the relative clause subjects and verbs, even when these were close together. (For example, it might predict _The boys who the girl *chase see the dog_, getting the number agreement of _boys_ and _see_ right, but failing on the more proximal—and presumably, easier—_girl chase_**s**.)

This failure, of course, is exactly what might have been predicted by Chomsky, Miller, and Gold.

## Incremental input

In an attempt to understand where the breakdown was occurring, and just how complex a language the network might be able to learn, I next devised a regimen in which the training input was organized into corpora of increasing complexity, and the network was trained first with the simplest input. There were five phases in all. In the first phase, 10,000 sentences consisting solely of simple sentences were presented. The network was trained on five exposures ("epochs") to this database. At the conclusion of this phase, the training data were discarded and the network was exposed to a new set of sentences. In this second phase, 7,500 of the sentences were simple, and 2,500 complex sentences were also included. As before, the network was trained for 5 epochs, after which performance was also quite high, even on the complex sentences. In phase three, the mixture was 5,000 simple/5,000 complex sentences, for 5 epochs. In phase four, the mixture was 2,500 simple/7,500 complex. And in phase five, the network was trained on 10,000 complex sentences.

Since the prediction task—given this grammar—is non-deterministic, the best measure of performance is not the extent to which the literal prediction is correct (measured thus, an error of 0.0 would require that the network memorize the training data) but rather the degree to which the network's predictions match the conditional probability distributions of the training data. Performance using this metric was very good at the conclusion of all phases of training, including the final phase. Final performance yielded an error of 0.177, with network output measured against the empirically derived likelihood estimates. (Alternatively, one can measure the cosine of the angle between these two vectors. Mean cosine at the end of training was 0.852; perfect performance would have been 1.00.) Furthermore, the network's high performance generalized to a variety of novel sentences which systematically test the capacity to predict grammatically correct forms across a range of different structures.

This result contrasts strikingly with the earlier failure of the network to learn when the full corpus was presented at the outset.[1] Put simply, the network was unable to learn the complex grammar when trained from the outset with the full "adult" language. However, when the training data were selected such that simple sentences were presented first, the network succeeded not only mastering in these, but then going on to master the complex sentences as well.

In one sense, this is a pleasing result, because the behavior of the network partially resembles that of children. Children do not begin by mastering the adult language in all its complexity. Rather, they begin with the simplest of structures, and build incrementally until they achieve the adult language.

There is an important disanalogy, however, between the way in which the network was trained and the way children learn language. In this simulation, the network was placed in an environment which was carefully constructed so that it only encountered the simple sentences at the beginning. As learning and performance progressed, the environment was gradually enriched by the inclusion of more and more complex sentences. But this is not a good model for the situation in which children learn language. Although there is evidence that adults modify their language to some extent when interacting with children, it is not clear that these modifications affect the grammatical structure of the adult speech. Unlike the network, children hear exemplars of all aspects of the adult language from the beginning.

If it is not true that the child's environment changes radically (as in this first simulation), what is true is that the *child* changes during the period he or she is learning language. A more realistic network model would have a constant learning environment, but some aspect of the network itself would undergo change during learning.

[1]. Both this result and the earlier failure were replicated several times with different starting conditions, a variety of different architectures, and various settings of the learning parameters (learning rate, momentum, bounds on beginning random weight initialization).

## Incremental memory

One developmental change which is plausibly relevant to learning is the gradual increase in memory and attention span which is characteristic of children (Kail, 1984). In the network, the analog of memory is supplied by the access the network has (via the recurrent connections) to its own prior internal states. The network can be given a more limited memory by depriving it of access, periodically, to this feedback. The network would thus have only a limited temporal window within which patterns could be processed.

A second simulation was therefore carried out with the goal of seeing what the effect would be, not of staging the input, but of beginning with a limited memory and gradually increasing memory span. The rationale was that this scenario more closely resembles the conditions under which children learn language.

In this simulation, the network was trained from the outset with the full adult language (i.e., the target corpus that had previously been shown to be unlearnable when it was presented from the beginning). However, the network itself was modified such that during the first phase, the recurrent feedback was eliminated after every third or fourth word (randomly).[1] In the second phase, the network continued with another set of sentences drawn from the adult language (the first set was discarded simply so the network would not be able to memorize it); more importantly, the memory window was increased to 4-5 words. In the third phase, the memory window was increased to 5-6 words; in the fourth phase, to 6-7 words; and in the fifth phase, the feedback was not interfered with at all.

Under these conditions, it turned out that the first phase had to be extended to much longer than in the previous simulation in order to achieve a comparable level of performance (12 epochs rather than 5; for purposes of comparison, performance was measured only on the simple sentences even though the network was trained on complex sentences as well). However, once this initially prolonged stage of learning was over, learning proceeded quickly through the remaining stages (5 epochs per stage). At the end, performance on both the training data, and also on a wide range of novel data, was as good as in the prior simulation. If the learning mechanism itself was allowed to undergo “maturational changes” (in this case, increasing its memory capacity) during learning, then outcome was just as good as if the environment itself had been gradually complicated.

Before discussing some of the implications of this finding, it is important to try to understand exactly what the basic mechanism is which results in the apparently paradoxical finding that learning can be improved under conditions of limited capacity. One would like to know, for example, whether this outcome is always to be expected, or whether this result might be obtained in only special circumstances.

We begin by looking at the way the network eventually solved the problem of representing complex sentences. The network has available to it, in the form of its hidden unit patterns, a high-

[1] This was done by setting the context units to values of 0.5.

dimensional space for internal representations. It is well known that in such networks these internal representations can play a key role in the solution to a problem. Among other things, the internal representations permit the network to escape the tyranny of a form-based interpretation of the world. Sometimes the *form* of an input is not a reliable indicator of how it should be treated. Put another way, appearances can deceive. In such cases, the network uses its hidden units to construct a *functionally-based* representational scheme. Thus, the similarity structure of the internal representations can be more reliable indicator of “meaning” than the similarity structure of the bare inputs.

In this simulation, the network utilized the various dimensions of the internal state to represent a number of different factors which were relevant to the task. These include: individual lexical item; grammatical category (noun, verb, relative pronoun, etc.); number (singular vs. plural); grammatical role (subject vs. object); level of embedding (main clause, subordinate, etc.); and verb argument type (transitive, intransitive, optional). Principle component analysis (Gonzalez & Wintz, 1977) can be used to identify the specific dimensions associated with each factor. The internal representations of specific sentences can then be visualized as movements through this state space (one looks at selected dimensions or planes, chosen to illustrate the factor of interest). Example trajectories for several sentences are shown in Figures 2-4.

*— Insert Figures 2-4 about here —*

One might think of such plots as the network equivalent of graphs of EEG activity recorded from human subjects while they process various types of sentences. Figure 2, for example, shows how the singular/plural distinction of the main clause subject is encoded and preserved during an embedded relative clause. Figure 3 shows how differences in verb-argument structure are encoded (in this grammar, *chases* requires a direct object, *sees* optionally permits one, and *walks* is intransitive). Figure 4 demonstrates the way in which the network represents embedded relative clauses.

One can also visualize the representational space more globally by having the network process a large number of sentences, and recording the positions in state space for each word; and then displaying the overall positions. This is done in Figure 5a. Three dimensions (out of the 70 total) are shown; the *x* and *y* coordinates together encode depth of embedding and the *z* coordinate encodes number (see Elman, 1991, for details).

*— Insert Figure 5 about here —*

At the outset of learning, of course, none of these dimensions have been assigned to these functions. If one passes the same sentences through a network prior to training, the internal representations have no discernible structure. These internal representations are the important outcome of learning; they are also the necessary basis for good performance.

The state-space graph shown in Figure 5a was produced under conditions of incremental training, which, we have seen, was crucial for successful learning. What does the state-space look like under conditions of failure, such as when we train a fully-mature network on the adult corpus from the beginning? Figure 5b shows such a plot.

Unlike Figure 5a, Figure 5b reveals a less clearly organized use of the state space. There is far greater variability, and words have noisier internal representations. We do not see the kind of sharp distinctions which are associated with the encoding of number, verb argument type, and embedding as we do when the network has succeeded in mastering the language. Why might this be?

When the network is confronted from the beginning with the entire adult corpus the problem is this. There are actually a relatively small number of sources of variance: number, grammatical category, verb-argument type, and level of embedding. However, these sources of variance interact in complex ways. Some of the interactions involve fairly long-distance dependencies. For example, in the (difficult to understand) sentence *The girl who the dogs that I chased down the block frightened, ran away*, the evidence that the verb *frightened* is transitive is a bit obscure, because the direct object (*the girl*) not only does not occur after the verb (the normal position for a direct object in simple English sentences), but occurs 10 words earlier; and there are several other nouns and verbs in between. Like people, simple recurrent networks do not have perfect memory. The network is able to find a solution to the task which works enough of the time to yield reasonable performance, but the solution is imperfect and results in a set of internal representations which do not reflect the true underlying sources of variance.

When learning proceeds in an incremental fashion, either because the environment has been altered or because the network itself is initially handicapped, the result is that the network only sees a subset of the data. When the input is staged, the data are just the simple sentences. When the network is given a limited temporal window, the data are the full adult language but the *effective* data are only those sentences, and portions of sentences, which fall within the window. These are the simple sentences. (Now we see why the initial phase of learning takes a bit longer in this condition; the network also has to wade through a great deal of input which is essentially noise.)

This subset of data, the simple sentences, contain only three of the four sources of variance (grammatical category, number, and verb argument type) and there are no long-distance dependencies. As a result, the network is able to develop internal representations which encode these sources of variance. When learning advances (either because of new input, or because improvements in the network's memory capacity give it a larger temporal window), all additional changes are constrained by this early commitment to the basic grammatical factors.

The effect of early learning, thus, is to constrain the solution space to a much smaller region. The solution space is initially very large, and contains many false solutions (in network parlance, local error minima). The chances of stumbling on the correct solution are small. However, by selectively focusing on the simpler set of facts, the network appears to learn the basic distinctions—noun/verb/relative pronoun, singular/plural, etc.—which form the necessary basis for learning the more difficult set of facts which arise with complex sentences.

Seen in this light, the early limitations on memory capacity assume a more positive character. One might have predicted that the more powerful the network, the greater its ability to learn a complex domain. However, this appears not always to be the case. If the domain is of sufficient complexity, and if there are abundant false solutions, then the opportunities for failure are great. What is required is some way to artificially constrain the solution space to just that region which contains the true solution. The initial memory limitations fill this role; they act as a filter on the input, and focus learning on just that subset of facts which lay the foundation for future success.

## PART II: How networks learn

We turn now to an intriguing problem. Answering that problem will require that we seek a deeper understanding of the principles which constrain learning in networks of the sort we have studied here, and ways in which network learning may differ from more classical learning systems.

The problem is this. We have just seen that there are conditions where a network appears to do better at learning a problem when it begins with a restricted subset of the data. This is the starting small result. However, we also know that there are conditions under which starting small can be disastrous; restricting the training data in such cases results in the network's learning the wrong generalization.

A simple example of this is the exclusive-OR function (XOR). This is a Boolean function of two inputs. When the inputs are identical the function maps to false (or 0); when the inputs are different, the function maps to true (or 1):

| INPUT | OUTPUT |
|---|---|
| 1 0 | 1 |
| 0 1 | 1 |
| 0 0 | 0 |
| 1 1 | 0 |

This function cannot be learned by two-layer networks with sigmoidal activation functions, but require at least one additional intermediate layer. Even then, the function is not always learned successfully. It is particularly important that the network see all the patterns from the outset. If the fourth pattern (for example) is withheld until late in training, the network will typically fail to learn XOR. Instead, it will learn logical OR, since this is compatible with the first three patterns. Worse, having learned OR, the network will be unable to modify its weights in a way which accommodates the final pattern. Harris (1991) has shown that similar results hold for the parity function (of which XOR is a reduced case). In general, experience with neural networks suggests that these systems not only thrive on training data but may require large data sets in order to learn a difficult function.

These results appear to be at sharp variance with the results presented earlier in this paper. However, both effects arise as a consequence of fundamental properties of learning in connectionist models.[1] I would therefore now like to consider what some of these properties might be, how they might differ from other approaches to learning, and what relevance they might have to under-

[1] To be precise, it should be made clear that the properties to be discussed here are associated with a specific approach to learning (gradient descent) and a specific class of connectionist networks (those involving distributed representations over populations of units with continuously valued non-linear activation functions). A variety of other architectures and learning algorithms have been studied in the literature. The principles proposed here do not necessarily extend to those approaches.

standing the interaction between learning and development in humans.

There are four properties I will focus on here, although there are a number of others which are relevant and have been discussed elsewhere (e.g., Bates & Elman, 1992; McClelland, in press). The properties I consider here are: (1) the statistical basis for learning and the problem of small sample size; (2) the representation of experience; (3) constraints on new hypotheses; and (4) how the ability to learn changes over time. Each property imposes a small constraint on learning but taken together the four characteristics sharply limit the power of networks. As we shall see, the effect of embedding learning in a system which develops over time (*i.e.*, starts small) is to exactly compensate for these limitations.

### Property 1: Statistics as the basis for learning; the problem of sample size.

In most neural network learning algorithms, the driving force for inference is statistics. The nature of statistical inference is a complex topic and the importance of statistics for learning in neural networks has engendered a certain amount of controversy.

To a large extent, conclusions about the inadequacy of statistically-based learning arise from claims advanced in connection with language learning. In a well-known paper, Miller and Chomsky (1963) argued that certain properties of natural language make statistically-based learning infeasible. The problem is exemplified in sentences such as *The people who say they want to rent your house next summer while you are away in Europe are from California.* Note that there exists a dependency between the number (plural) of *people* early in the sentence, and the number (plural) of the second occurrence of the verb *are*, 17 words later. Let us suppose that a learner is confronted with the task of determining the conditions under which *are*, rather than *is*, should be used. Miller and Chomsky argued that if the learner is able only to use cooccurrence statistics, then an inordinate number of sentences will have to be sampled. This is because if the dependency is viewed as a matter of statistical cooccurrence rather than as a structural fact relating to subject-verb agreement in sentences (which may contain embedded relative clauses), the learner will have to have previously sampled all the possible sentences which contain *people* and *are* separated by all possible 17-word combinations. This number is astronomical (it actually outstrips the number of seconds in an individual’s lifetime by many orders of magnitude). Miller and Chomsky concluded that statistically-based learning is therefore inadequate to account for language acquisition.

However, it is important to distinguish between the use of statistics as the *driving force* for learning, and statistics as the *outcome* of learning. If the learning mechanism is merely compiling a lookup table of cooccurrence facts (*i.e.*, statistics as output), then the approach is indeed doomed. But this is not what neural networks do, and presumably not what humans do either. Neural networks are function approximators, not compilers of lookup tables. The goal of learning is to discover the function which underlies the training data. The learning algorithm is statistically driven and is highly sensitive to the statistics of the training data. The outcome of learning, however, is rather closer to what Miller and Chomsky would have called a rule system than it is to a lookup table of statistics. Practically, this means that networks are able to extrapolate beyond their training data in ways which obviates the need (for example) to see all possible combinations of words in sentences. In other words, networks generalize.

There *is* a problem associated with statistical-based learning, however, which turns out to be relevant to us here. This is the problem which arises when statistics are computed over small sample sizes. In general, the smaller the sample size (N), the riskier it is that the sample statistics provide a good estimate of the population statistics. With small N, there may be a large number of reasonable generalizations which are compatible with the data at hand; as N grows, new data will typically exclude some of these generalizations. (In principle, there are always an infinite number of generalizations which are compatible with any data set of any given size; but in practice, the effect of additional data is to more highly constrain the number "reasonable" generalizations.)

The case of XOR cited earlier is one example of this. Or consider the following data in which patterns are classified into one of two categories (0 or 1).

| *PATTERN* | *CLASSIFICATION* |
|---|---|
| 1 0 1 1 0 1 | 1 |
| 0 0 0 0 0 0 | 1 |
| 0 0 1 1 0 0 | 1 |
| 0 1 0 1 1 0 | 0 |
| 1 1 1 0 1 1 | 0 |
| 0 0 0 1 1 1 | 0 |

Let us assume that a network is trained on these data. We then present it with a novel pattern,

0 1 1 1 0 1

The question is, will this be classified as a member of class 0 or class 1?

The answer depends on which generalization the network has extracted from the training data. There are multiple possibilities consistent with the limited observations. The network might have discovered that the first three patterns (in class 1) are symmetrical about the center (the last three bits are the mirror image of the first three), in which case the network will assign the test pattern to class 0 (because it is nonsymmetrical). Alternatively, the network might have discovered that all members of class 1 have even parity, and so it will classify the test pattern as class 1 (because it has even parity). Or the network might have extracted the generalization that all members of class 0 have a 1 in the fifth bit position, while class 1 patterns have a 0 in that position. In this case the test item belongs in class 1. (And of course, because the outcome is the same as symmetry, we cannot know with this test item whether the network is making the classification on the basis of symmetry or contents of fifth bit position; further probes would be needed.) Thus, the effect of limited data is to impose minimal constraints on the nature of the generalizations possible. Increasing the data set may restrict the range of generalizations which the network can extract.

Why should this be a problem for neural networks? Certainly at early stages of learning there may be a limited amount of training data, but why should the problem not disappear with continued exposure to new data? To understand why the problem persists, we need to consider the remaining three properties of learning.

### Property 2: The representation of experience.

Given the importance of data to all models of learning, a very basic question arises about "lifetime" of the data and about the form in which prior training examples are stored. This question is rarely addressed in any explicit way. In many models of learning (e.g., Dunbar & Klahr, 1989; Lea & Simon, 1979; Osherson, Stob, & Weinstein, 1986) it seems to be assumed that the data accumulate and are available in a more or less veridical form for as long as they may be needed. Whenever a current hypothesis is rejected or found to be lacking, a new hypothesis (or modification) must be generated, based on the old data plus whatever new information prompts the revision. So it would seem that the data must be preserved for as long as they are needed, and must be maintained in a more or less unprocessed form. Exemplar-based models make this assumption quite explicitly (Estes, 1986; Medin & Schaffer, 1978; Nosofsky, in press) and claim that experiences are stored as individually retrievable exemplars.

Connectionist models of the sort described here make very different assumptions regarding the lifetime and representational form of training examples. When a network is presented with a training pattern, the learning algorithm results in small changes in the connection strengths (weights) between nodes. These weights implement the function the network has learned up to that point. Once a given pattern has been processed and the network has been updated, the data disappear.[1] Their effect is immediate and results in a modification of the knowledge state the of the network. The data persist only implicitly by virtue of the effect they have on what the network knows. The data themselves are lost and are not available to the learning algorithm for later re-processing (for example, in a way which might allow the learning mechanism to generate alternative hypotheses). This leads us to the next property of learning in which has to do with constraints on the generation of new hypotheses.

### Property 3: Constraints on new hypotheses: The continuity of search.

Consider the space of all possible hypotheses that a system might entertain. In traditional learning models, these hypotheses usually take the form of symbolic propositions or rule sets. The space of possible hypotheses for such a system consists of all rules which conform to the grammar of the system. In connectionist models, on the other hand, hypotheses are implemented as values of the weights on connections between nodes.

Now consider the trajectory, over time, of the search in the two spaces. In the traditional system, this trajectory may be continuous (*e.g.,* through the gradual introduction, deletion, or change of rules), but it need not be. That is, successive hypotheses need not be particularly similar to one another. When one hypothesis is discarded, the succeeding hypothesis may differ wildly from it. In part, this follows as a consequence of having some faithful and enduring record of the prior evidence (see Property 2). The evidence may be rearranged in novel ways which are unconstrained by the temporal history of learning.

[1] (For convenience, many simulations randomly re-cycle the same patterns at later points in training, but this is not to be confused with the internal storage and explicit manipulation of data. The important point is that the data themselves are not individually represented in the network.)

In neural networks employing gradient descent, the picture is quite different. The search through hypothesis space is necessarily continuous. To make this somewhat clearer, and to introduce some concepts which will be useful in later discussion, imagine the simple network shown at the top of Figure 6. The network has only one input and two nodes (one hidden, and one output); there are two weights (one from the input, and one from hidden to output).

*— Insert Figure 6 about here —*

Below the network, we have shown a hypothetical graph of the error that might be produced (for some hypothetical data set, which we will not specify here) as we systematically vary values of the two weights. The different values of the two weights are shown along the X and Y axes, and the error which would be produced by the network at the different possible weight values is shown along the Z axis. (By error we mean the discrepancy between what the network would output in response to the training data, compared with the correct output for those data.) If we knew this error surface in advance, of course, we could set our network to the combination of weights which produces the lowest error (marked as point *d* in the figure). Not knowing what this surface looks like, we might determine it empirically by systematically sweeping through all possible combinations of weights and testing the network at each point. This is of course quite tedious (particularly given networks with a larger number of weights). What gradient descent learning algorithms provide us are techniques for exploring the error surface in an efficient manner which hopefully allows us to determine the combination of weights yielding the minimum error.

We begin with weights that are chosen randomly, often from a uniform distributed between +/- 1.0 so that we begin near the centroid (the importance of this will become apparent later). A possible starting point is shown in Figure 6 as point *a*. As data are processed, the learning algorithm lets us make small adjustments in our current weights in a way which leads to lower error (i.e., we follow the error gradient). Our goal is to proceed in this manner until we find the global error minimum (point *d*). Whether or not we succeed, or get trapped in a local minimum (point *e*; any small change in weights will increase the error, even though the current error is non-zero) depends on a number of factors, including how big are the steps through the weight space which we allow ourselves, as well as what the shape of the error surface looks like (because the error surface is a joint function of the network architecture and the problem at hand).

We will return shortly to some of these issues, but for present purposes we note that the nature of the network's hypothesis testing has a qualitatively different character than that in traditional systems. As previously noted, these latter approaches to learning permit a succession of radically different hypothesis to be entertained. The network, on the other hand, begins with some randomly chosen hypothesis (the initial weight settings) and is allowed to make small incremental changes in those settings. If we plot the trajectory of weights settings explored by gradient descent, we might see something which looks like the curve in Figure 6. New hypotheses are required to be similar to old hypotheses—but note that any two very similar hypotheses may differ dramatically in the output they produce (compare the error at points *b* and *c*). Thus, similar hypothesis may give rise to very different behaviors. But the important point here is that the gradient descent approach to learning imposes a constraint which prevents the network from generating wildly different hypotheses from one moment to the next. Learning occurs through smooth and small changes in hypotheses. Unfortunately, if the network falls into a local minimum, as at point *e*, this constraint may prevent it from escaping, dooming it forever to believe the wrong hypothesis.

We turn now to a final property, which once again constrains the nature of learning in networks.

### Property 4: How the ability to learn changes over time: Early flexibility vs. late rigidity.

The networks described here use backpropagation of error as a learning algorithm. This algorithm permits us to modify the weights in a network in response to the errors which are produced on training data. In most general terms, the algorithm can be understood as a way to do credit/blame assignment. Somewhat more specifically, the change involves the following weight adjustment equation.

$$\Delta w_{ij} = \eta \delta_i a_j$$

This equation says that the weight change between any two units *i* and *j* (where *i* indexes the receiver unit and *j* the sender unit) is the product of three terms. The first term, $\eta$, is a scaling constant and is referred to as the learning rate; it is typically a small value so that learning occurs in small increments. (Consider what might happen in Figure 6 if we begin at point *a* and make a very large weight change; we might oscillate forever between points *a* and *e*, missing the terrain in between.) The last term, $a_j$ is the activation of the sender, and implements the credit/blame aspect of the algorithm. The middle term, $\delta_i$, is the one I wish to focus on here. It is calculated as

$$\delta_i = f'(net)error$$

where ***error*** (in the case of an output unit) simply represents the discrepancy between the target output of the unit and the actual output, and ***f´(net)*** is the derivative of the receiver unit's activation function, given its current ***net*** input. The activation function used in most networks is sigmoidal, as shown in Figure 7.

*— Insert Figure 7 about here —*

This activation function has several important properties: (1) all input is "squashed" so that the unit's resulting activation lies between 0.0 and 1.0; (2) net input of 0.0 results in an activation of 0.5, which is in the middle of the unit's activation range; positive inputs result in activations greater than 0.5 and negative inputs yield activations less than 0.5; and (c) the activation function is monotonic but non-linear. The range of greatest sensitivity is around 0.0 input; the node's response saturates at large magnitude inputs (positive or negative).

This activation function has several interesting consequences as far as learning is concerned. Recall that it is customary to initialize networks to small random values around 0.0, lying near the centroid of the weight space (*e.g*., the region near the center of the space in Figure 6). This means that at the outset of learning, the net input to a node will typically be close to 0.0 (because the small negative and positive weights act as multipliers on the inputs, and since they are randomly determined with mean of 0.0, they tend to cancel each other out). Since net inputs close to 0.0 lie in the range of a unit's greatest sensitivity, at the outset of learning, nodes are activated in the region where they are most sensitive.

Secondly, we see that the derivative (or slope) of the activation function shown in Figure 7 is greatest in the mid-range. Near both extremes, the slope diminishes asymptotically toward 0.0. Recalling now that the actual weight change is a product of three terms, with one containing the slope of the activation function (for the given input), we see that weight changes will tend to decrease as a unit's activation saturates. This is true regardless of the unit's actual error. (A large error multiplied times a vanishingly small slope will still be a small number.)[1]

Since at early stages of learning the input to units tends to be in the mid-range, the consequence of all of this is that not only are units most sensitive to input during the onset of learning, but they are almost most easily modified. As learning progresses, the weights to a network tend to grow and the net input increases. This brings the units' activation into a range where they are less sensitive to small differences in input and leads to a behavior which tends to be more categorical. The earlier malleability also gives way to an increasing rigidity, such that the network is able to respond more effectively to mistakes early during learning and less so as learning progresses. Note this "ossification" is not the result of an independent process of maturation, but rather the direct result of learning itself. The more the system knows (whether right or wrong), the harder it is to learn something new.

• • •

We have considered four properties of learning in connectionist networks. We have seen that each of these properties in some way constrains or limits the ability of networks to learn. Summarizing the main conclusions:

(1) Networks rely on the representativeness of their data sets. With small sample size, a network may not discover the generalization which characterizes the larger population. This problem will be most serious at the early stages of learning since the sample size is necessarily smaller then.

(2) Networks are also most sensitive during the early period of learning. As learning progresses networks are less likely to be able to modify their weights. Taken together with the first observation, the network is *most* inclined to use information at a point in learning (the early stage) when that information may be *least* reliable.

[1] If cross-entropy is used an error measure rather than mean-squared error, this problem is avoided for output units; but it still occurs for non-output units.

(3) Gradient descent learning makes it difficult for a network to make dramatic changes in its hypotheses. Once a network is committed to an erroneous generalization it may be unable to escape this local minimum. Taken together with the second observation, the problem gets worse as learning proceeds.

The picture that emerges is of a system which is highly constrained, and in which the outcome of learning may be far from optimal. Indeed, it seems like a recipe for disaster. The approach differs markedly with other models of learning, in which—at least in principle, given deterministic inputs—"a 100% success rate can...be achieved on the basis of information available to the learner" (Estes, 1986). Networks can fail to learn a task for any of the reasons described above. They are far from perfect learners.

If this were all that one could say, the story would not be terribly interesting. But in fact, there are several strategies which may ameliorate these limitations. One can "arrange" to have better initial data; or one can "arrange" to have worse initial data. Oddly, both work.

The incremental learning strategy employed in the simulations described here is an example of how a system can learn a complex domain by having better initial data. The language problem is hard for the network to learn because crucial primitive notions (such as lexical category, subject/verb agreement, *etc*.) are obscured by the complex grammatical structures. This makes it difficult to learn the primitive representations. But here we have a Catch-22 problem: The network is also unable to learn about the complex grammatical structures because it lacks the primitive representations necessary to encode them. These difficulties are compounded by the network's early commitment to erroneous hypotheses, and its tendency to ossify over time. Incremental learning solves the problem by presenting the network with just the right data (*i.e.*, data which permit the network to learn the basic representational categories) at just the right time (*i.e.*, early on, when the network's plasticity is the greatest). A key aspect to the solution, as far as its possible relevance to the human case, is that there is a natural mechanism available for doing the filtering. By starting with an immature and impoverished memory which allows the system to process only simple sentences, the network constructs a scaffolding for later learning. As time progresses, the gradual improvement in memory capacity selects more and more complex sentences for processing.

Interestingly, exactly the opposite strategy can be employed: Arrange to have worse initial data. This can happen if the data are noisier at the outset of learning than later on.[1] The network's learning capacity is greatest at early stages, but this is also the time when its training data are most limited, and so the network runs the risk of committing itself to the wrong generalization. If the initial data are corrupted by noise, on the other hand, the increased variability may retard learning and keep the network in a state of flux until it has enough data to make reasonable approximations at the true generalization. Note that both effects may be achieved through the same mechanism, a developmental schedule in which initial capacity is reduced relative to the mature state.

With this perspective, the limited capacity of infants assumes a positive value. Limited ca-

[1] This might be introduced, for example, by adding random Gaussian noise to stimulus patterns.

pacity acts like a protective veil, shielding the infant from stimuli which may either be irrelevant or require prior learning to be interpreted. Limited capacity reduces the search space, so that the young learner may be able to entertain a small number of hypotheses about the world. And the noisiness of the immature nervous system may encourage generalizations which require a larger sample size.

Is there any empirical evidence in support of such a positive interaction between maturational limitations and language learning? Elissa Newport has suggested that indeed, early resource limitations might explain the apparent critical period during which languages can be learned with native-like proficiency. Newport calls this the “less is more” hypothesis (Newport, 1988, 1990). It is well-known that late learners of a language (either first or second) exhibit poorer performance, relative to early or native learner. Newport suggests that examination of the performance of early (or native) learners when it is at a comparable level to that of the late learners (*i.e.,* early on, while they are still learning) is particularly revealing. Although gross error scores may be similar, the nature of the errors made by the two groups differs. Late learners tend to have incomplete control of morphology and rely more heavily on fixed forms in which internal morphological elements are frozen in place and therefore often used inappropriately. Young native learners, in contrast, commit errors of omission more frequently.

Newport suggests that these differences are based in a differential ability to analyze the compositional structure of utterances, with younger language learners at an advantage. This occurs for two reasons. Newport points out that the combinatorics of learning the form-meaning mappings which underlie morphology are considerable, and grow exponentially with the number of forms and meanings. If one supposes that the younger learner is handicapped with a reduced short-term memory, then this reduces the search space, because the child will be able to perceive and store a limited number of forms. The adult’s greater storage and computational skills actually work to their disadvantage. Secondly, Newport hypothesizes that there is a close correspondence between perceptually salient units and morphologically relevant segmentation. With limited processing ability, one might expect children to be more attentive to this relationship than adults, who might be less attentive to perceptual cues and more inclined to rely on computational analysis. Newport’s conclusions are thus very similar to what is suggested by the network performance: There are situations in which maturational constraints play a positive role in learning. Counterintuitively, some problems simply can only be solved if you start small. Precocity is not always to be desired.

Turkewitz and Kenny (1982) have also argued that developmental limitations may not only be adaptive for an individual’s current state (*e.g.,* the immature motor system of a newborn animal prevents it from wandering away from its mother), but may also assist in neurogenesis and provide a basis for later perceptual development. For example, consider the problem of size constancy. This would seem to be a major prerequisite to an infant’s maintaining order in a world in which the projected size of an object may change dramatically with its distance from the infant. Turkewitz and Kenny suggest that the problem of learning size constancy may be made much easier by the fact that initially the infant’s depth of field is restricted to objects which are very close. The means that the problem of size constancy effectively does not arise at this stage. During this period, therefore, the infant is able to learn the relative size of objects in the absence of size constancy. This knowledge might then make it possible to learn about size constancy itself. (Consistent with this hypothesis is the observation that when size constancy comes in, around four to six months, it develops first for objects which are close; McKenzie, Tootell, & Day, 1980.)

This leads us to what is perhaps not an intuitively obvious perspective on development in biological systems, and in particular, on the value of early limitations. It is tempting to view early stages of development in a negative light. Infancy is sometimes seen as a period which must be somehow “gotten through.” And certainly, there are negative consequences to having sensorimotor, perceptual, and cognitive systems which are not fully developed. An individual that cannot perceive threats nor flee in the face of danger must be at an adaptive disadvantage; and the caretaking requirements that are imposed on the adults of the species limit their activities and consume considerable time and energy. So it is therefore surprising that evolutionary forces have *not* selected for individuals who are born fully functional. Perhaps counter-intuitively, the more complex the species, the greater the tendency for long periods of infancy. Humans are an extreme example, in which a significant fraction of an individual’s lifetime is spent in childhood.

One common explanation why prolonged infancy might not be selected out in humans is that, although maladaptive in itself, infancy is a compromise between two other traits—increased brain size and upright posture—which separately have significant adaptive value but are at odds with each other. For example, there is a positive advantage to upright posture, in that it frees two of the limbs for manipulative purposes. For biomechanical reasons, however, upright posture tends to force a narrowing of the pelvis, which in females also leads to a constriction in the birth canal. But this is at cross-purposes with the larger cranium associated with higher primates. To solve this problem, females have evolved to have a slightly wider pelvic girdle than males, which partially accommodates the larger brain cases which must pass through the birth canal. But also, the cranium in the infant is reduced in size relative to the adult cranium. The price paid—a longer period of immaturity—may have negative consequences, but these are outweighed by the positive advantages conferred by the ability to walk upright and to have a larger adult brain.

The present results do not provide any reason to reject such a hypothesis. What the current findings suggest are rather an additional reason why a lengthy period of development may be adaptive. Although I have focussed here on limitations on learning, the kind of learning mechanism afforded by a neural network is actually quite powerful, and there are good reasons why such a mechanism might be favored by evolution. But it is not a perfect system. It is subject to the various limitations described above. In addition, the environment in which humans function is itself highly complex, and some of the domains (e.g., language) may have a large number of “false solutions” (i.e., solutions which fit the examples but do not lead to correct generalizations). The shortcomings of this style of learning are neatly compensated by the limitations present during early childhood.

More generally, the lesson this suggests is that if we wish to understand either developmental phenomena or aspects of human learning, it is important to study the ways they interact. In isolation, we see that both learning and prolonged development have characteristics which appear to be undesirable. Working together, they result in a combination which is highly adaptive.

## Acknowledgments

I am grateful to Elizabeth Bates, Cathy Harris, Mark Johnson, Annette Karmiloff-Smith, Virginia Marchman, Jay McClelland, and Domenico Parisi for many stimulating discussions about the issues discussed in this paper. This work was supported by a contract from Army Avionics (Ft. Monmouth), and by a grant from the John D. and Catherine T. MacArthur Foundation.

## References

Ash, T. (1989). Dynamic node creation in backpropagation networks. *Connection Science*, 1, 365-375.

Baker, C.L. (1979). Syntactic theory and the projection problem. *Linguistic Inquiry*, 10:533-581.

Bowerman, M. (1987). The 'no negative evidence' problem: How do children avoid constructing an overly general grammar? In J.A. Hawkins (Ed.), *Explaining language universals.* Oxford: Basil Blackwell.

Braine, M.D.S. (1971). On two types of models of the internalization of grammars. In D.I. Slobin (Ed.), *The ontogenesis of grammar: A theoretical perspective.* New York: Academic Press.

Brown, R., & Hanlon, C. (1970). Derivational complexity and order of acquisition in child speech. In J.R. Hayes (Ed.), *Cognition and the development of language.* New York: Wiley.

Chomsky, N. (1957). *Syntactic structures.* The Hague: Mouton.

Dunbar, K. & Klahr, D. (1989). Developmental differences in scientific discovery processes. In D. Klahr and K. Kotovsky (Eds.), *The 21st Carnegie-Mellon symposium on cognition: Complex information processing: The Impact of Herbert A. Simon.* Hillsdale, NJ: Lawrence Erlbaum.

Elman, J.L. (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.

Estes, W.K. (1986). Array models for category learning. *Cognitive Psychology,* 18:500-549.

Fahlman, S.E., & Lebiere, C. (1990). The Cascade-Correlation learning architecture. In D.S. Touretzky (Ed.), *Advances in Neural Information Processing Systems 2*, 524-532.

Gold, E.M. (1967). Language identification in the limit. *Information and Control*, 16:447-474.

Gonzalez, R.C., & Wintz, P. (1977). *Digital image processing.* Reading, MA: Addison-Wesley.

Harris, C. (1991). *Parallel distributed processing models and metaphors for language and development.* Ph.D. dissertation, University of California, San Diego.

Jordan, M. I. (1986). Serial order: A parallel distributed processing approach. Institute for Cognitive Science Report 8604. University of California, San Diego.

Kail, R. (1984). *The development of memory.* New York: W.H. Freeman.

Lea, G. & Simon, H.A. (1979). Problem solving and rule induction. In H.A. Simon (Ed.), *Models of Thought.* New Haven, CT: Yale University Press.

McClelland, J.L. (in press). Parallel distributed processing: Implications for cognition and development. In R. Morris (Ed.), Parallel distributed processing: Implications for psychology and neurobiology. Oxford: Oxford University Press.

McKenzie, B.E., Tootell, H.E., & Day, R.H. (1980). Development of visual size constancy during the first year of human infancy. *Developmental Psychology*, 16:163-174.

Medin, D.L., & Schaffer, M.M. (1978). Context theory of classification learning. *Psychological Review,* 85:207-238.

Miller, G.A., & Chomsky, N. (1963). Finitary models of language users. In R.D. Luce, R.R. Bush, & E. Galanter (Eds.), *Handbook of Mathematical Psychology, Volume II.* New York: John Wiley.

Newport, E.L. (1988). Constraints on learning and their role in language acquisition: Studies of the acquisition of American Sign Language. *Language Sciences,* 10:147-172.

Newport, E.L. (1990). Maturational constraints on language learning. *Cognitive Science,* 14:11-28.

Nosofsky, R.M. (in press). Exemplars, prototypes, and similarity rules. In A. Healy, S. Kosslyn, & R. Shiffrin (Eds.), *From learning theory to connectionist theory: Essays in honor of William K. Estes* (Vol. 1). Hillsdale, NJ: Lawrence Erlbaum.

Osherson, D.N., Stob, M., & Weinstein, S. (1986). *Systems that learn: An introduction to learning theory for cognitive and computer scientists.* Cambridge, MA: MIT Press.

Pinker, S. (1989). *Learnability and cognition.* Cambridge, MA: MIT Press.

Plunkett, K., & Marchman, V. (1990). From rote learning to system building. Center for Research in Language, TR 9020. University of California, San Diego.

Pollack, J.B. (1990). Language acquisition via strange automata. *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum.

Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning internal representations by error propagation. In D.E. Rumelhart & J.L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition (Vol. 1)*. Cambridge, MA: MIT Press.

Servan-Schreiber, D., Cleeremans, A., & McClelland, J.L. (1986). Encoding sequential structure in simple recurrent networks. CMU Technical Report CMU-CS-88-183. Computer Science Department, Carnegie-Mellon University.

Shultz, T.R., & Schmidt, W.C. (1991). A Cascade-Correlation model of balance scale phenomenon. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum.

Turkewitz, G., & Kenny, P.A. (1982). Limitations on input as a basis for neural organization and perceptual development: A preliminary theoretical statement. *Developmental Psychobiology,* 15(4):257-368.

Wexler, K., & Cullicover, P. (1980) *Formal principles of language acquisition.* Cambridge, MA: MIT Press.

## Figure Legends

1. Schematic of the simple recurrent network used in the simulations. Rectangles incidate blocks of units; the number of units in each block is indicated by the side. Forward connections (dotted lines) are trainable; the solid downward connections from Hidden to Context units are fixed at 1.0, and link units on a one-to-one basis.
2. Plot of the movement through one dimension of the hidden unit activation space (the second principal component) as the successfully trained network processes the sentences *boy who boys chase chases boy* vs. *boys who boys chase chase boy*. The second principal component encodes the singular/plural distinction in the main clause subject.
3. Plot of the movement through two dimensions of hidden unit activation space (first and third principal components) as the network processes the sentences *boy chases boy*, *boy sees boy*, and *boy walks* (sentence endings are indicated with ]S). Nouns occupy the right portion of this plane, and verbs occupy the left side; the axis running from top left to bottom right encodes the verb argument expectations.
4. Plot of the movement through two dimensions of hidden unit activation space (first and eleventh principal components) as the network processes the sentences *boy chases boy* and *boy who chases boy who chases boy*. The depth of embedding is encoded by a shift to the left in the trajectory for the canonical simple sentences.
5. (a) Sample of points visited in the hidden unit state space of a successfully trained network as it processes 1,000 randomly chosen sentences. (b) Sample of points visited in the hidden unit state space of a network which has failed to learn the task, as it processes 1,000 randomly chosen sentences.
6. Hypothetical error surface (bottom) associated with a network with two trainable weights (top). The *z* coordinate indicates the error that is produced if the network has weights corresponding to the values at the *x* and *y* coordinates; low regions in this surface correspond to low error.
7. The logistic function used to determine node activations. One term in the weight change equation is the slope of this function; when node activations saturate at either 1.0 or 0.0, this slope asympotically approaches 0.

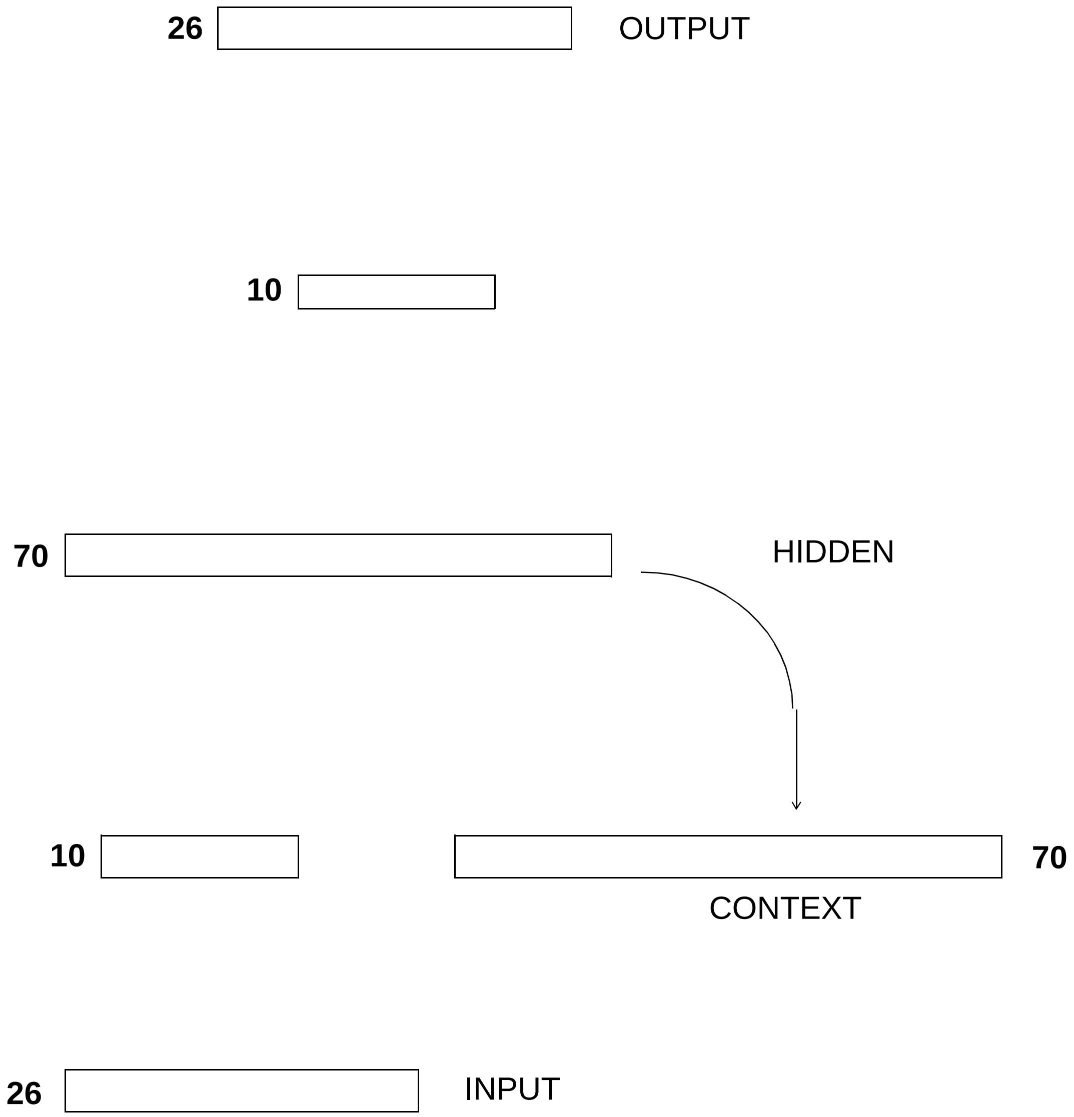
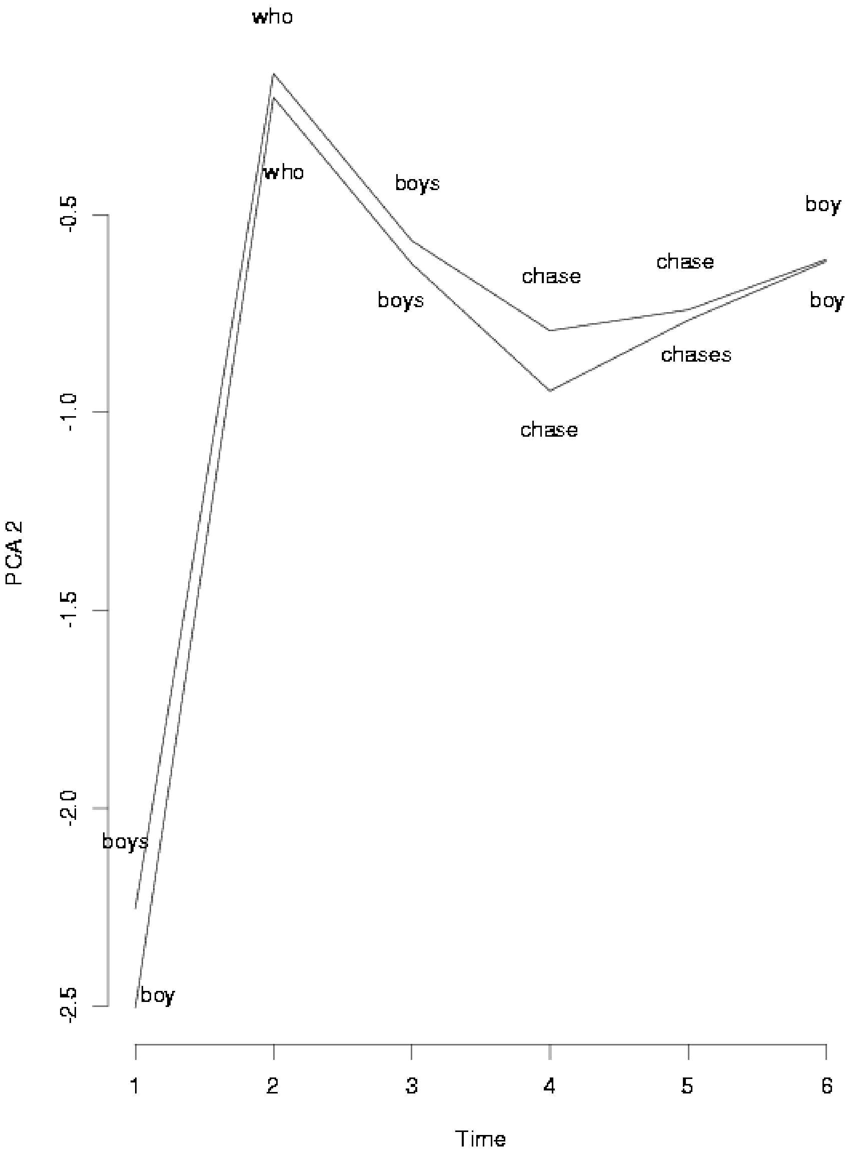
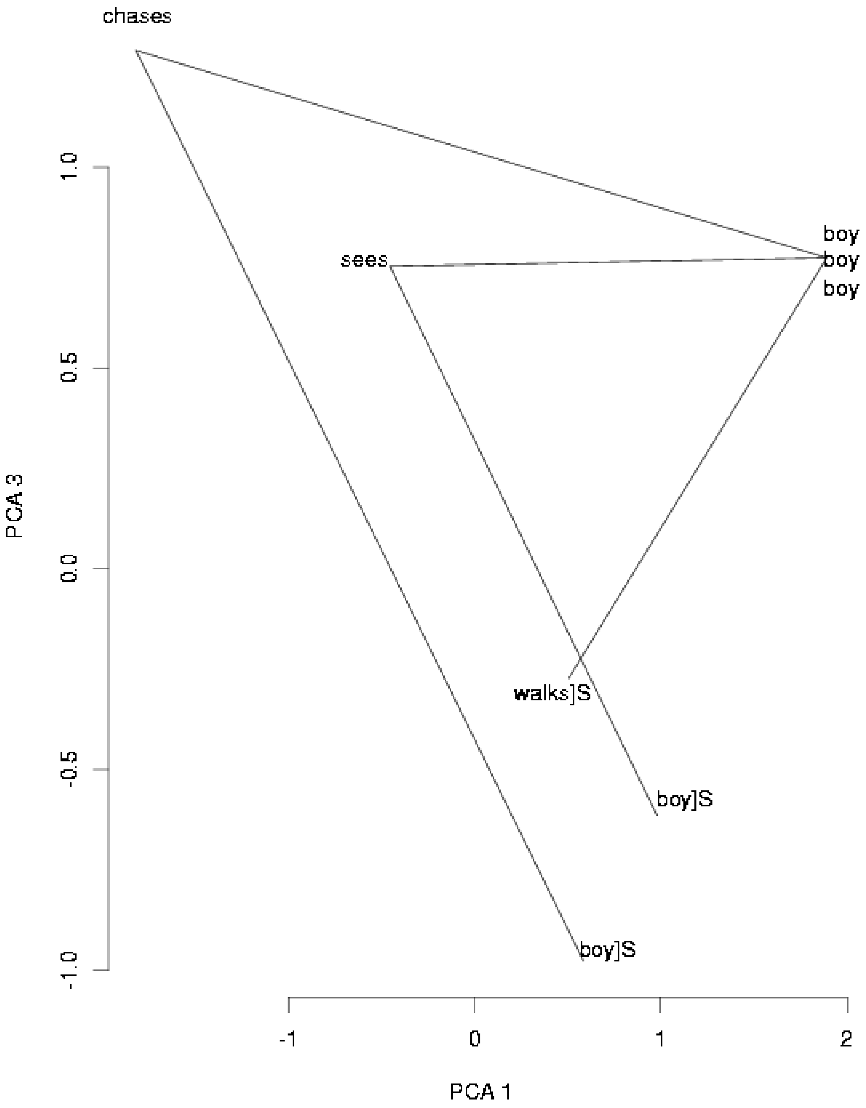
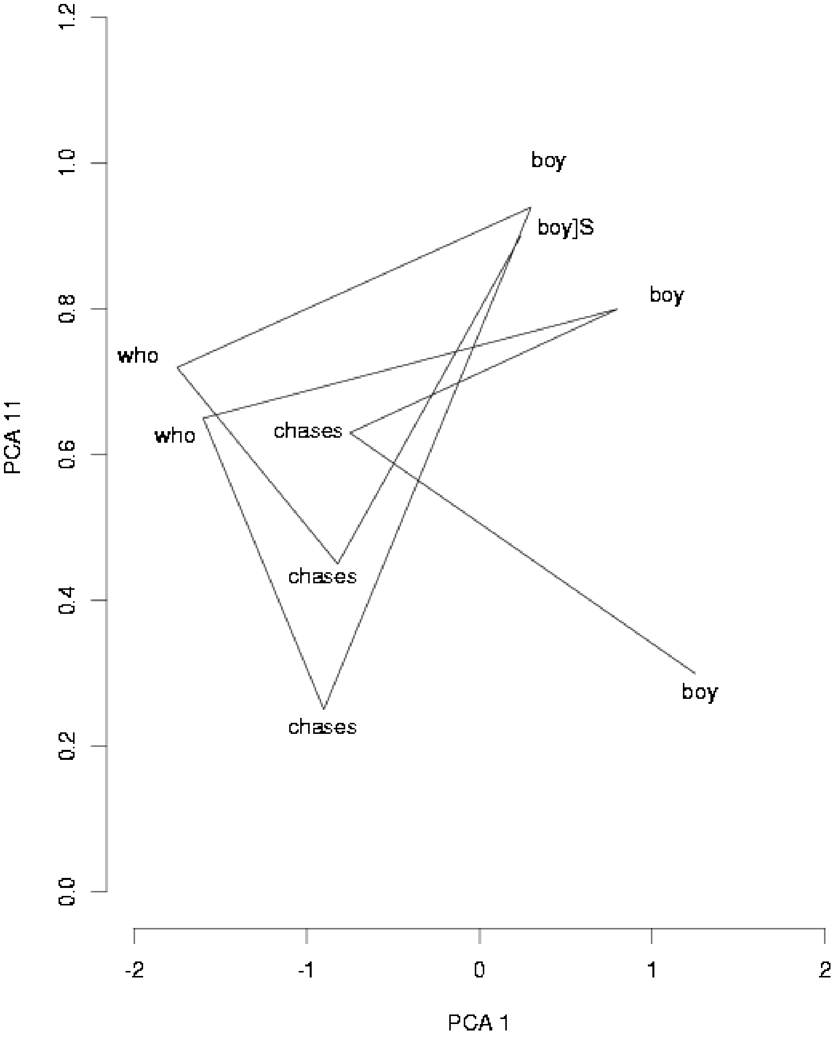
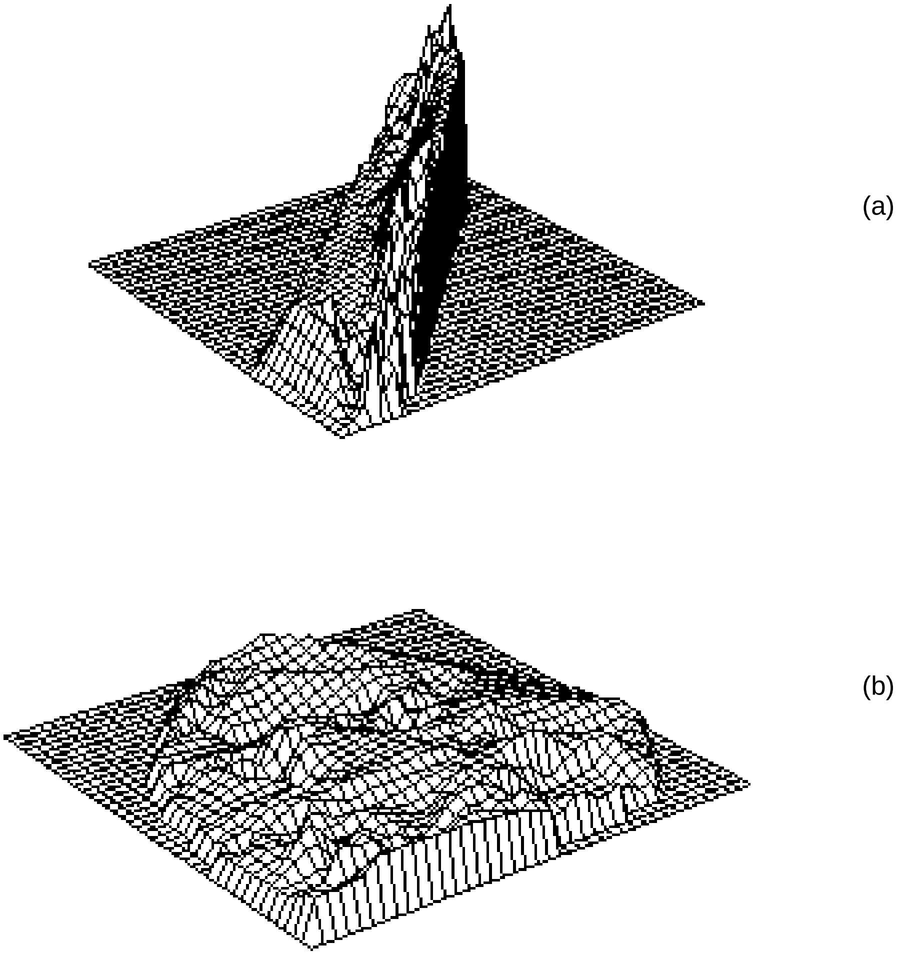


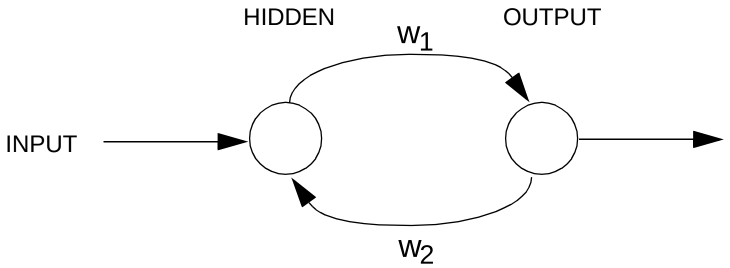
Figure 1

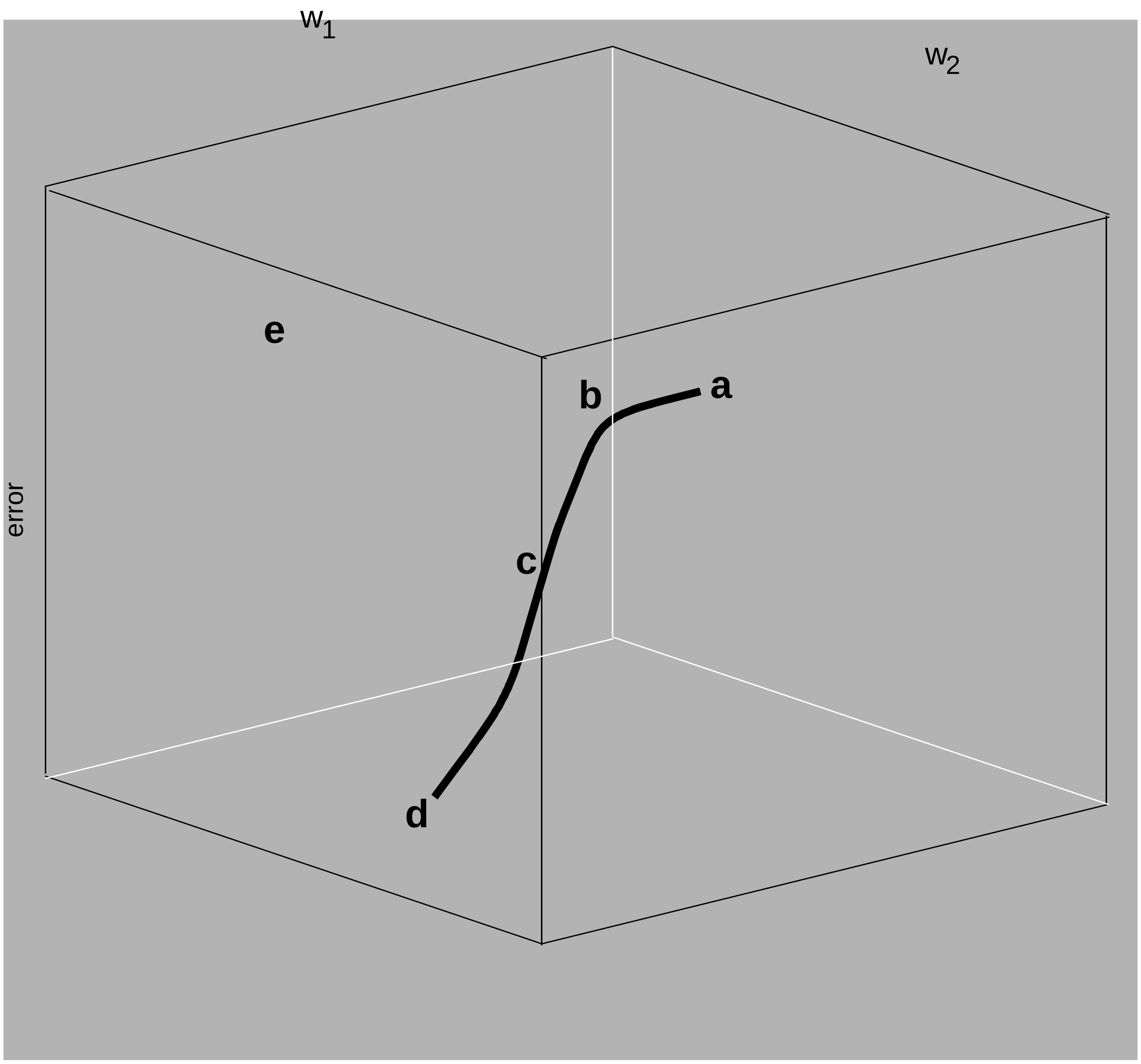

who
who
boys
boys
chase
chase
chase
chases
boy
boy
boys
boy
-0.5
-1.0
-1.5
-2.0
-2.5
PCA 2
1
2
3
4
5
6
Time

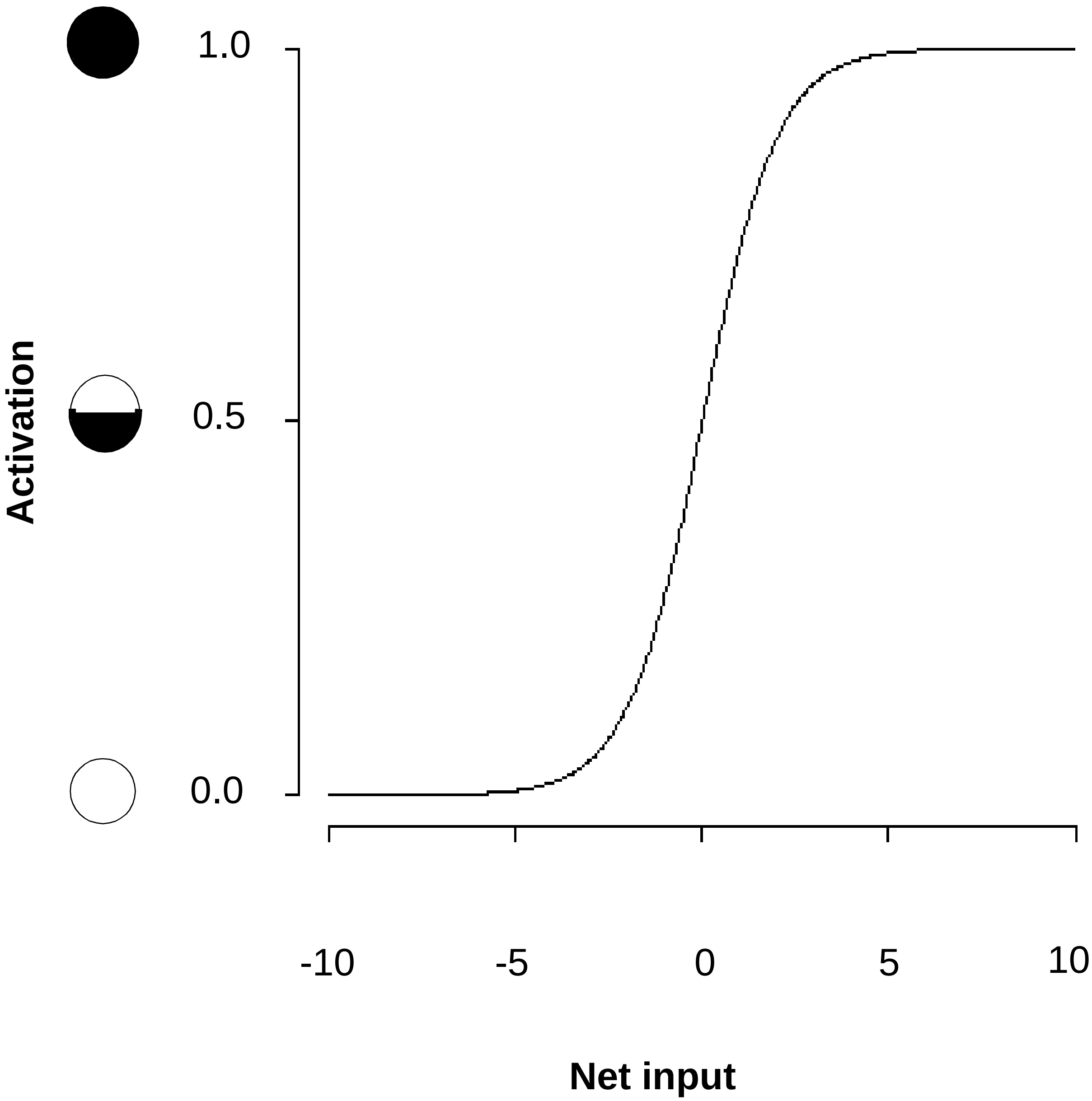

chases
sees
boy
boy
boy
walks]S
boy]S
boy]S
PCA 3
1.0
0.5
0.0
-0.5
-1.0
-1
0
1
2
PCA 1


boy
boy]S
boy
who
who
chases
chases
chases
boy
PCA 11
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-2
-1
0
1
2
PCA 1


(a)
(b)


HIDDEN
$w_1$
OUTPUT
INPUT
$w_2$


$w_1$
$w_2$
e
b
a
error
c
d


1.0
0.5
0.0
Activation
-10
-5
0
5
10
Net input