UC Merced

Proceedings of the Annual Meeting of the Cognitive Science Society

Title

Using Connectionist Networks to Examine thue Role of Prior Constraints in Human Learning

Permalink

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 16(0)

Authors

Harm, Michael
Altmann, Lori
Seidenberg, Mark S.

Publication Date

1994

Peer reviewed

Using Connectionist Networks to Examine the
Role of Prior Constraints in Human Learning

Michael Harm
Computer Science Department
University of Southern California
mharm@gizmo.usc.edu

Lori Altmann
Linguistics Department
University of Southern California
lori@gizmo.usc.edu

Mark S. Seidenberg
Neuroscience Program
University of Southern California
marks@neuro.usc.edu

Abstract

This research investigated the effects of prior knowledge on learning in psychologically-plausible connectionist networks. This issue was examined with respect to the benchmark orthography-to-phonology mapping task (Sejnowski & Rosenberg, 1986; Seidenberg & McClelland, 1989). Learning about the correspondences between orthography and phonology is a critical step in learning to read. Children (unlike the networks mentioned above) bring to this task extensive knowledge about the sound-structure of their language. We first describe a simple neural network that acquired some of this phonological knowledge. We then summarize simulations showing that having this knowledge in place facilitates the acquisition of orthographic-phonological correspondences, producing a higher level of asymptotic performance with fewer implausible errors and better nonword generalization. The results suggest that connectionist networks may provide closer approximations to human performance if they incorporate more realistic assumptions about relevant sorts of background knowledge.

Introduction

Although cognitive scientists have emphasized how children's acquisition of knowledge is constrained by prior knowledge (either innate or the result of prior learning), such constraints are rarely incorporated in connectionist networks. Consider, for example, the well-studied task of learning the correspondences between orthography (spelling) and phonology (sound). This knowledge plays an important role in learning to read (Adams, 1989); moreover, the inconsistencies in these correspondences in English (FIVE-GIVE, HERE-WERE, etc.) present an interesting learning problem. Sejnowski and Rosenberg's NETtalk (1987) was the first connectionist model applied to this task; Seidenberg and McClelland (1989) developed a related model that simulated detailed aspects of human performance in reading words aloud. The SM89 model was limited in two important respects, however. First, it performed more poorly than people on generalization trials (reading nonwords such as JINJE or KEDE) (Besner et al, 1990). Second, many of the pronunciations that it produced as errors contained phoneme sequences that are not permitted in English (e.g., SPIN --> SIPN). The defects in this model have been taken as reflecting important limitations on the capacities of neural networks to capture detailed aspects of human behavior (e.g., Coltheart et al., 1993; McCloskey, 1991; Prasada & Pinker, 1993).

In this paper we examine an important difference between these connectionist networks and children who are learning to read that may account for some of these discrepancies. Beginning readers already possess extensive knowledge of the structure of spoken language. They have learned which phonemic segments occur in their language and about the constraints that govern phoneme sequencing (the phonotactics of the language). Their task is then to learn how orthographic symbols relate to this phonological structure. The Seidenberg and McClelland model, in contrast, possessed no prior knowledge of phonology and was initialized with random weights. This created a more difficult learning task than the one confronting the beginning reader: the network had to learn about phonological structure at the same time it was learning to map orthography onto it.

We examined the role of prior knowledge in such networks in two steps. We first developed a connectionist network that learned about the phonological structure of English monosyllables. We then examined how including or excluding this phonological knowledge affected the task of learning orthographic-phonological correspondences. We compared two networks that were identical except that one was configured with the phonological structure that was learned in the first simulation, and the other was configured with the usual random weights. Results suggest that providing prior knowledge of phonological structure results in more rapid learning, a greatly reduced tendency to produce implausible utterances, and better nonword generalization.

Simulation 1: Induction Of Phonological Constraints

The network consisted of 29 units that were fully connected to each other, plus the bias associated with each unit. The 29 units were used to encode words consisting of CVC sequences of phonemes. Phonemes were represented in terms of standard phonetic features (e.g., voiced, labial); each unit corresponded to one of these features; 12 feature bits were used for each consonant and 5 for the vowel. Weights on connections between units were initially randomized and each unit's connection to itself was frozen at 0.5. The effect of this is that a unit's activation,

independent of all other inputs, slowly drops off from its initial value over time.

The training set consisted of a set of 564 CVC words taken from an online dictionary. The training procedure was as follows. The probability that a word would be selected for training was a function of its Kucera and Francis (1967) frequency. A word was selected and at timestep 0, all 29 units were clamped with its correct phonological representation. At timesteps 1 to 4, they were unclamped and allowed to mutually activate and de-activate each other. A unit's aggregate input is first computed as the weighted sum of the current output of all other units connected to it. This aggregate input is applied to a sigmoidal squashing function. Formally, the output o of unit j at time t is

$$o_j(t) = f\left(\sum_i^I W_{i,j} \bullet o_i(t-1) \right)$$ where I is the set of units connected to

unit j, $W_{i,j}$ is the weight from unit i to unit j, and f is the sigmoid function.

The output at each time step from 1 to 4 was compared with the input phonemes. Where the output disagreed with the phonemic representation, a sum of squared error signal was computed. Thus, the network was being asked to recreate and hold the pattern that had been present at step 0 over steps 1-4. The weights connecting the units to each other were modified according to the standard backprop through time algorithm for training recurrent connectionist nets (Williams & Zipser, 1989, 1990). Because each unit's auto-connection was frozen, each unit needed input from its neighbors to hold its former value. A unit received a high error signal when it failed to receive sufficient activation or inhibition from its neighbors; thus, the learning algorithm causes agonistic or antagonistic tendencies among the units to be represented in the weight space. Specifically, the model encoded both the intra-segmental regularities (i.e., the fact that only some combinations of features produce actual phonemes) and intersegmental regularities (i.e., the fact that only some sequences of phonemes are legal). The resulting weight space represents a set of stable attractor states (Plaut & McClelland, 1993). These states represent phonemically and phonotactically legal sound patterns in the target language represented by the CVCs.

We evaluated the effectiveness of the representations formed in this stage by determining the extent to which the network was able to perform pattern completion. Given a partially-specified input, could the network use knowledge of phonological structure to generate legal patterns? A sample of items from the training corpus of all CVC words was prepared. Each form was presented into a matrix of weights taken from the prewiring network. For each output unit in the form, we determined whether the unit was getting the correct level of activation from its neighbors. The unit being evaluated was unclamped, and the summed input to that unit from its neighbors (i.e. the dot product of the feature values and the weight vector) was compared with the unit's real value for that form. Thus the phonological output space was disturbed by the deletion of one unit, and the test determined to what extent that one unit could be pulled into its correct value by the activations of its neighbors.

If the summed input to a unit from its neighbors was negative, and the unit was supposed to be off, or if the summed input was positive and the unit was supposed to be on, the unit was scored as receiving correct activation from its neighbors. If not, it was incorrect. The average squared value of the units' correct activation from its neighbors minus the squared value of incorrect activations is multiplied by the frequency of the word form. This gave a scalar measure of the accuracy to which the weights encoded regularities in the representation for that unit for that word form. This test was done over all units in the phonological representation, and the resulting score for each unit was then multiplied by the frequency of the word form being tested. This way, an error made within the word form CZAR did not penalize the network as much as an error in the word form CAR would. This test was repeated for each word form in the training corpus, and the recorded results for each unit were averaged. This gave a measure of the probability that the network's weight space could coerce unspecified values into a legal pattern.
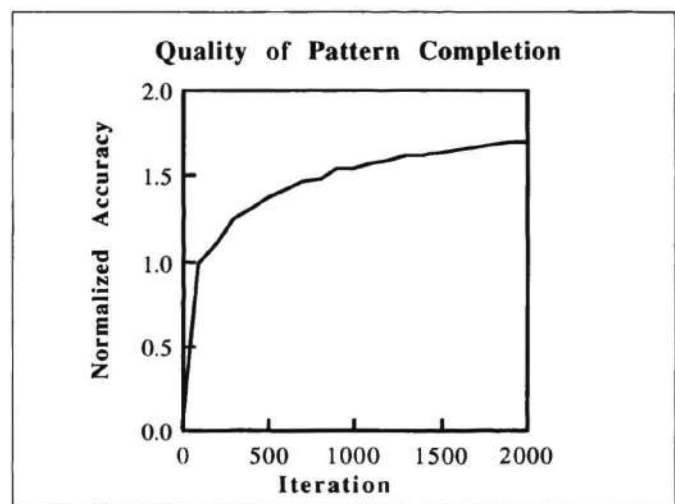

Figure 1

This measure was evaluated during the training of the net, and its value is shown in figure 1. Initially, with the weights randomly assigned, the value was zero; a unit is no more likely to receive correct activation from its neighbors than it is to get incorrect activation. By the end of 2000 training epochs, a weighted average of 83% of the output units across all CVC words receive correct activation from their neighbors. Thus, the model had encoded sufficient information about phonological structure to tolerate degradation of the input pattern. This knowledge is not perfect, but neither is the child's at the onset of reading education.

Simulations 2-3: The Reading Task

The network used in these simulations was constructed as follows. The output layer consisted of the same 29 units representing phonological output as in Simulation 1; there also were 80 hidden units and 27 orthographic input units. Input units were connected to all hidden and output units. The hidden units were fully connected to each other, and to the output units. Similarly, the output units were fully

connected to each other and to all hidden units. Biases were connected to all non-input units.

Training Regime

A set of input-output pairs was prepared. The input forms consisted of orthographic strings corresponding to the spelling of the words from Simulation 1. The output forms were the feature values for the phonological representations of the words. The network was recurrent, and had to hold the output value over time in a stable state.

Letters were presented serially over the 27 input units, which provided a localist encoding of the alphabet, with an additional unit indicating whether the input is a null character or not. A null character was represented by turning units 1-26 off and unit 27 on. Thus for the word CAT, the network would activate the unit corresponding to C at time 0, A at time 1, and T at time 2. All other units are turned off (output a value of -1). Each unit in the network computes its next value as the weighted sum of the current outputs of all units that are connected to it.

Upon presentation of the final character, an error signal is computed, which is the squared difference between the target value for that unit and the unit's actual output value at that time step. This error is used to accumulate changes to the weight matrix W again using backprop through time.

For two time steps following the presentation of the last character, a null character is presented to the input units of the network, and input is propagated through the net as before. Error signals are computed for these last two time steps in the same way; the squared difference between the target output and the actual output at those time steps is calculated, and again this error signal is used to update the weight matrix. Thus for a word of k letters, we clamped the input units with the representation of character t during time steps $t=[0,k-1]$, and a null character for time steps $t=[k,k+1]$. Error was injected and propagated through the network to update the weight matrix during time steps $t=[k-1,k+1]$.

Presenting the letters serially allowed the network to capture phonological regularities (e.g., that T can be followed by H but not the reverse). Ideally, the network input pattern of TH should develop a state transition within the hidden units that has common components across words like THIS, OTHER, etc (Plaut & McClelland, 1993). Such commonalties are difficult to represent in the slot-based orthographic representations used in many other models (e.g., Daugherty & Seidenberg, 1992).

Testing Procedure

Three networks with identical architectures, training sets, and training regimens but different initial weights were evaluated. All weights in the three networks were first randomized to values distributed normally between -1 and 1. For the structured network (SN), the weights between all of the phonological output units (and their respective biases) were initialized to the values that were the outcome of Simulation 1. These weights were scaled down by a constant factor, due to the greater fan-in of the orthographic task network. For the unstructured network (UN), the randomly generated set of weights were retained. The standard deviation of these two sets of weights differed,

owing to the tendency of the Simulation 1 net to push weights to extreme positive or negative values. In order to ensure that this difference between the networks was not the cause of any observed differences in performance, a third, control network (CN) was set up, using weights that were a random permutation of those found in the structured network.

Results

Figure 2 compares the three reading task networks in terms of their ability to learn the training set. The structured network learned the target pronunciations faster than the other two, and reached a higher asymptotic level of performance as well. Figure 3 shows the number of utterances produced by the three networks containing nonexistent phonemes (i.e., illegal combinations of features). The structured network was much better at producing well-formed (though sometimes incorrect) output patterns.
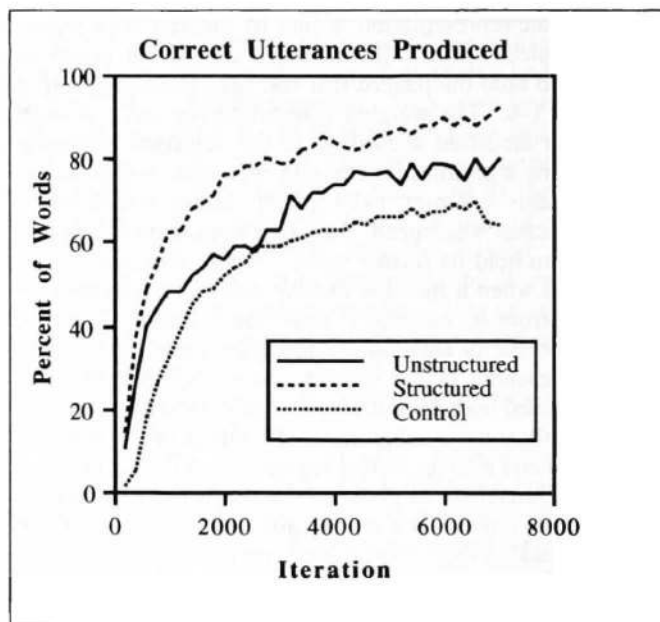

Figure 2

Figure 4 shows a histogram of errors for the structured and randomized networks, broken down by which segment caused the error. 74% of the errors made by the structured net occur in the vowel, with only 14% occurring within the first consonant and the remainder in the final consonant. Further, the majority of the errors within the vowel segment were caused by a single feature. In contrast, the unstructured network made 46% of its errors in the initial consonant, 42% in the vowel, and 12% in the trailing consonant. The unstructured network has a much wider distribution of error types, while the structured network exhibits more systematic errors.

A set of 53 nonwords was used to assess the generalization performance of the structured and unstructured networks. Nonword pronunciation is difficult to assess because human subjects often generate multiple pronunciations, whereas the model only produces one
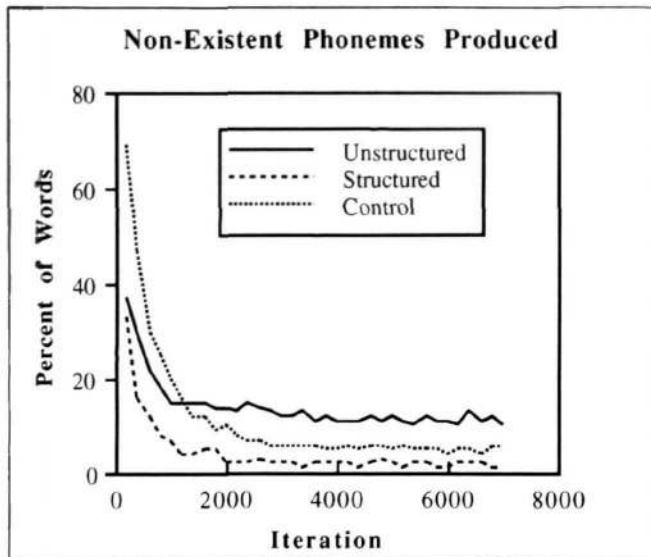
Non-Existent Phonemes Produced

Figure 3

(Seidenberg et al, 1994). Overall, the structured and unstructured networks generated approximately the same number of incorrect utterances. However, the unstructured network produced far more words containing illegal phonological segments (9/53) than the structured network (2/53). Further, in the nonword test, both networks followed a pattern very similar to that seen in the training set: the structured network's errors were focused on the vowels, while the unstructured network exhibited a wider distribution of error types, being divided almost equally between errors in the first consonant and errors in the vowel.
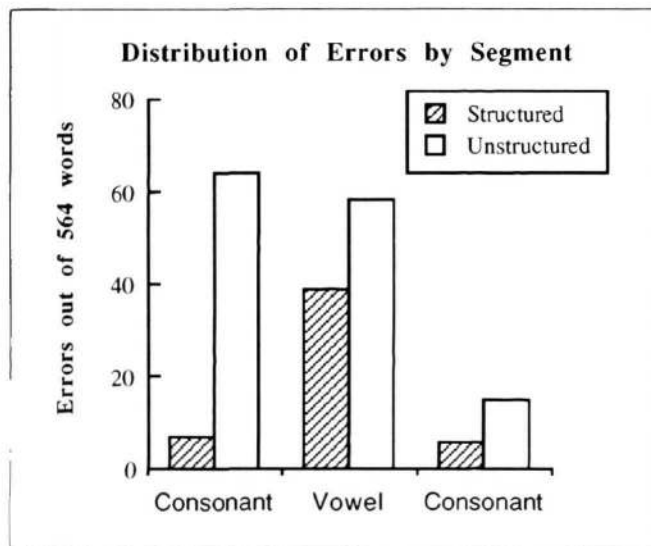
Distribution of Errors by Segment

Figure 4

Conclusions

In summary, a simple recurrent backpropagation network performed better on the task of mastering the correspondences between spelling and pronunciation when it was provided with prior information about phonological structure than when it was not. Prestructuring the network did not provide direct information about orthographic-phonological correspondences; rather, it provided constraints on the structure of target phonological patterns. This allowed the limited resources that were available to be focused on the main problem of learning orthographic-phonological correspondences. The child who is learning to read is comparable to a prestructured network in the sense that he/she already possesses extensive knowledge of the sound structure of language. In fact, there is good evidence that success in learning to read is related to preliterate phonological knowledge (Gough et al., 1992). Children who perform well on so-called "phonological awareness" tasks, such as deciding if two words rhyme or deleting a sound from a word ("say SPLIT without the P"), acquire early word decoding skills more rapidly. The simulations provide a simple explanation for why this would occur.

More generally, the simulations suggest that connectionist networks can provide better simulations of human behavior by being more realistic about the state of people's knowledge at the onset of learning. Connectionist models tend to rely on the brute force power of the learning algorithm to encode generalizations in a problem domain. By ignoring relevant pre-existing knowledge, these networks create learning problems that are more difficult than necessary. This may account in part for the relatively poor performance of some connectionist networks compared to people. This also contributes to the impression that the connectionist approach is incompatible with the existence of a priori constraints. The present research suggests instead that the approach provides a way to explore the role of biological and other types of constraints (Seidenberg, 1992).

The simulations we have described are limited in several respects that should be acknowledged. The purpose of the simulations was to examine the role of constraints on a simple kind of learning. The networks that were used do not represent general solutions to the spelling-sound mapping problem, which entails other issues not addressed here. Thus, for example, even the prestructured model's nonword performance was not as good as that reported by Plaut et al. (1994). This is due to factors such as the limited size of the training corpus that were not immediately relevant to the constraint issue. Much better performance could be achieved using extensions of the networks we have described here, however. In addition, the simulations examined the effects of constraints that were derived from the pretraining experience, which represents a simplification of the situation confronting the beginning reader in an important sense. The child's knowledge of phonology is based in part on experience, i.e. exposure to a spoken language. The child must learn about the inventory of phonemes that the language happens to use and constraints on the order of phonemes. However, this learning is further constrained by human perceptual and motor capacities. Because of these biological constraints, only some phonemes and phoneme sequences are possible. We have not attempted to separate the effects of these innate constraints from the effects of prior experience with the language, but this is an obvious step for future research. In

395

this way connectionist models might contribute to understanding how different types of constraints influence human learning.

References

Adams, M. (1990). **Beginning to Read.** Cambridge, MA: MIT Press.

Coltheart, M., Curtis, B., Atkins, P., & Haller, M. (1993). Models of reading aloud: Dual-route and parallel distributed processing approaches. **Psychological Review,** 100, 589-608.

Daugherty, K., & Seidenberg, M.S. (1992). Rules or connections? The past tense revisited. **Proceedings of the 14th annual meeting of the Cognitive Science Society.** Hillsdale, NJ: Erlbaum.

Gough, P., Ehri, L., & Treiman, R. (1992). **Reading Acquisition.** Hillsdale, NJ: Erlbaum.

Kucera, H., & Francis, W.N. (1967). **Computational analysis of present-day American English.** Providence, RI: Brown University Press.

McClosky, M (1991). Networks and theories: The place of connectionism in cogntivie science. **Psychological Science,** 2, 387-395

Plaut, D., & McClelland, J.L. (1993). Generalization with componential attractors: Word and nonword reading in an attractor network. **Proceedings of the Cognitive Science Society.** Hillsdale, NJ: Erlbaum, pp. 824-829.

Prasada, S., & Pinker, S. (1993). Generalization of regular and irregular morphological patterns. **Language And Cognitive Processes,** 8, 1-56.

Seidenberg, M.S. (1992). Connectionism without tears. In S. Davis (Ed.), **Connectionism: Advances in Theory and Practice.** Oxford University Press.

Seidenberg, M. S., & McClelland, J. L. (1990). More words but still no lexicon: Reply to Besner et al. **Psychological Review, 97,** 447-452.

Seidenberg, M.S., Plaut, D.C., Petersen, A., McClelland, J.L., & McRae, K. ((in press)). Nonword generalization and models of word recognition. **Journal Of Experimental Psychology: Human Perception And Performance.**

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

Williams, R. J. and Zipser, D. 1989. A learning algorithm for continually running fully recurrent neural networks. **Neural Comp.** 1, 270-280

Williams, R. J. and Zipser, D. 1990. Gradient-based learning algorithms for recurrent connectionist networks. Tech Rep. NU-CCS-900-9. Northeastern University, College of Computer Science, Boston