

A Connectionist Perspective on Prosodic Structure

Author(s): Mary Hare, David Corina, and Garrison Cottrell

Proceedings of the Fifteenth Annual Meeting of the Berkeley Linguistics Society (1989), pp. 114-125

Please see “How to cite” in the online sidebar for full citation information.

Please contact BLS regarding any further use of this work. BLS retains copyright for both print and screen forms of the publication. BLS may be contacted via <http://linguistics.berkeley.edu/bls/>.

The Annual Proceedings of the Berkeley Linguistics Society is published online via [eLanguage](#), the Linguistic Society of America's digital publishing platform.

Mary Hare, David Corina, and Garrison Cottrell
University of California, San Diego

One area of interest in current phonological research is the representation and analysis of prosodic structure in language. This research is concerned not with the minimal contrastive units, or phonemes, in isolation, but with higher levels of organization which govern these units in context. Such research relies on representations such as the mora, syllable, or foot, and attempts to characterize well-formedness conditions on these representations. These constraints are often expressed as rules which maintain well-formed structures.

Accounts relying on prosodic structure in language are firmly entrenched in current linguistic theory and account for a wide variety of facts (McCarthy and Prince 1986, Clements and Keyser 1983, Hayes 1989). The current paper will begin by considering one such analysis, the syllable-based account of Vowel Epenthesis in Turkish, as discussed in Kornfilt (1986). This example demonstrates a standard linguistic treatment of prosodic phenomena in which segmental content is pressured to conform to well-formedness conditions imposed by higher-order prosodic units, in this case the syllable. We will then reconsider these data from the perspective of Parallel Distributed Processing models, with the goal of investigating the development of these higher-order constraints. In the Kornfilt account, the existence of the syllabic template is presupposed. Our question is whether such an assumption is necessary, or if the functional equivalent of well-formedness conditions may be abstracted from samples of positive data. In this paper we develop two implementations of a particular learning algorithm and demonstrate that functional constraints on well-formed outputs may indeed be induced from a limited number of correct examples.

I. Vowel Epenthesis

A formal treatment of Turkish vowel epenthesis is described below (Kornfilt 1986, Clements and Sezer 1982). This approach involves associating a word of Turkish with a template corresponding to the well-formed syllables of that language. If an abstract underlying form of the word does not correspond to the structure demanded by the template, it is not allowed to surface. Instead, some process (here, vowel epenthesis) occurs to "save" the otherwise illicit form.

The example we are concerned with here involves data shown in (1). Although here we give only a subset of the relevant data, there is reason to assume that the epenthesis account is to be preferred to an account involving deletion.

Notice that in (1), the high vowel in the second syllable of

the nominative form is absent in the accusative. Further, while the nominative is unaffixed, the accusative is formed by the addition of a high vowel.

(1)	nom	acc	
	fi.kir	fik.ri	'idea'
	a.kil	ak.li	'intelligence'

This account relies on the certain well-formedness constraints on syllable structure in Turkish. In this language, only three types of syllable-final consonant clusters are allowed:

- sonorant + obstruent (*türk*, 'Turk')
- voiceless fricative + oral stop (*çift*, 'double')
- k + s (*raks*, 'dance')

In the examples below, (2) shows the accusative form of the word, and the syllabification of the CV template with which it is associated. Notice that since the accusative form is V-final, this form divides correctly into two acceptable syllables, and no change is required. (3) and (4) show the derivation of the nominative or unaffixed form of the same root. In (3) the first three segments of the word comprise a licit syllable, but the final *r* cannot combine with the *k* in the coda since a CS cluster is not allowed in coda position. The *r* is left unattached. This is not a well-formed representation, and is unacceptable. The representation in (4), on the other hand, is well-formed. Here a second vowel has been epenthesized before the *r*, creating an acceptable syllable. The form in (3) is assumed to be the underlying representation associated with this word, and serves as input to the vowel epenthesis rule which results in the form shown in (4).

(2)	§	§	(3)	§	(4)	§	§
	/ \	/		/ \		/\	/ \
	C V C	C V		* C V C C		C V	C V C
	f i k	r i		f i k r		f i	k i r

The causal process in the examples above is a rule which is sensitive to syllabic representations and acts to preserve or create well-formed structures.

The remainder of this paper will take a different approach to these data. In the simulations described below, prosodic structure does not have the status of presupposed representations related by rules. Instead, generalizations describing correct output are developed in the course of producing legal samples of a given language. The implicit structure derived from this process then serves to constrain subsequent language production. The paper describes two Parallel Distributed Processing (PDP) networks

developed to model this view of prosodic structure as a dynamic process. Before turning to the models, we will provide a brief overview of Parallel Distributed Processing.

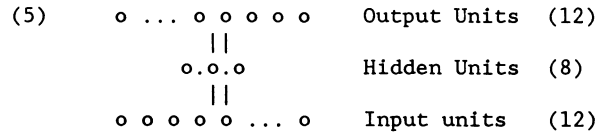
II. Parallel Distributed Processing

Parallel Distributed Processing (Rumelhart and McClelland 1986a, McClelland and Rumelhart 1986) is a computational theory of cognitive modeling inspired by neural information processing rather than by the traditional von Neumann computer metaphor. In these models, knowledge is represented in weighted connections that spread patterns of activation over large numbers of interconnected processing units. Networks of this type have been shown to be successful in a variety of constraint satisfaction tasks.

Although PDP networks may take a wide variety of forms, the two described in this paper consist of three levels of processing units: input units, which respond directly to stimuli from outside the system; output units, whose activation patterns represent the system's response to that input; and one or more levels of "hidden" or intermediate units. Further details concerning these models will be given during the description of the simulations.

III. Simulation I

The first model uses the architecture diagramed in the figure below.



This type of network is known as an auto-associator, or identity map. In an auto-associator, the input and output patterns are always the same. That is, given a certain input pattern, the task of the model is to reproduce that pattern as output. In order to understand why performance on such a simple task might lead to interesting consequences, it is necessary to understand certain principles about this type of architecture.

In models of this type an activation pattern is presented on a bank of input units. Activation is then sent from this input layer to a second layer of internal units, where the input pattern must be re-represented. This internal pattern is then mapped to the output units. In our model, as shown in figure (5), the input and output layers each consist of 12 processing units, while the internal layer has only 8. More patterns may be encoded on a 12-bit bank than on an 8-bit bank, yet in order to reach the output, the model must go through the step of re-encoding the input patterns on this smaller internal layer. Clearly, the inputs cannot be literally copied on the internal layer. The only way the model

can successfully complete the identity task is to discover what the most useful generalizations about the input patterns are, and use the internal layer to form an efficient representation of these generalizations or constituent subpatterns. In other words, the use of an auto-associator with a compressed internal layer is a way of forcing a network to abstract generalizations out of the input patterns. As a consequence of this architecture, a pattern may only be reproduced if the network is able to view it as some lawful combination of the constituent sub-patterns.

In the simulation to be described, we are interested in discovering whether a network, when exposed to examples of acceptable strings, can extract the prototypes corresponding to licit syllable templates. One way of knowing whether a network has accomplished this task is to present it with an ill-formed string, and see if it changes this string in ways which are consonant with well-formed templates. In the simulation described below we will use this task as a measure of the success of the induction process.

Our assumption has been that if a model is given the task of producing well-formed words, the sub-patterns taken as significant would be those which correspond to canonical syllable patterns, and the model will then be pressured into producing outputs which conform to those patterns. In testing this assumption we used data such as that shown in figure (6). Input was designed to represent well-formed Turkish words, or rather to correspond to a CV skeleton for those words, with no segmental content. For example, the Turkish words in (a) were represented in the model as the strings in (b).

(6)	(a) dik	(b) CVC
	sirk	CVSC
	boksit	CVCCVC

(were C = consonant, S = sonorant, V = vowel)

Each slot in the skeleton was represented by two processing units. One unit encoded information about segment type (C, V, or S) while the other gave information about position in the syllable. The coding scheme is given below.

(7) syllable position	segment type
1 = peak	1 = vowel
0 = onset	0 = consonant
.5 = coda	.75 = sonorant
or second half	
of long vowel	
.25 = sonorant in 2nd	
position in onset	

Thus the Turkish word *brut* received the following input

representation:

(8)	b	0	0
	r	.25	.75
	u	1	1
	t	.5	0

The input *brut* occupies eight of the twelve input units. The input was 12 units to allow for strings of up to six segments. When the string was shorter, as in this example, the remainder of the input space was padded with random entries.

There were 34 input patterns in all. The model was trained to reproduce the input on the output layer. This training is accomplished through a learning algorithm called back-propagation of error (Rumelhart, Hinton, and Williams 1986). Back propagation works in the following way: When an input is presented to the network, activation is propagated forward through the system, leading to some pattern of activation on the output layer. The first time an input is presented, output is essentially random. This actual output is compared to the target and the discrepancy between the two is computed. This discrepancy between the actual and the desired output is the error produced by the network on that pattern. The weights on the connections between the input, internal, and output units are then modified slightly in order to minimize this error. At this point the input pattern is presented again, and the process is repeated until the error meets some criterion of acceptability. This network was trained for 1000 epochs, where an epoch equals one presentation of each pattern in the training set. At this point the model responded perfectly to the training patterns.

The interesting question is not whether the model learned to produce the output correctly, but what generalizations about the data it had formed in the course of the learning. To determine this, we tested the network on 22 novel strings. Recall that the response of the model to novel data is conditioned by the generalizations that have been abstracted from the training set. If the model has extracted prototypes corresponding to licit syllable types, this will be clear in its response to new inputs. It will be able to reproduce inputs which can be viewed as lawful combinations of the correct subpatterns (i.e. prototypically good strings). However, when faced with input that deviates from these canonical patterns, the network will be unable to reproduce it. Instead, the network will force the output to conform as much as possible to a canonically good string.

The results of the test show this to be the case. The generalization test set was divided into three main classes. Ten items corresponded to prototypically acceptable strings in Turkish. A small number of test items, for example the string CCCV, were wildly unacceptable as examples of good syllable structure. The last three items were of intermediate acceptability, or have been posited to exist as underlying forms in the language.

In the first class, the model correctly reproduced 8 of the ten forms. The model responded in an interesting way to the other two items in this set. Both these items were patterns with complex onsets, as for example V.CSVC, where the second syllable begins with a consonant-sonorant cluster. The model made no clear decision on where to attach the first C, outputting a response ambiguous between V.CSVC and VC.SVC. This is striking in that although the model saw no examples of the second type, this in fact is also an acceptable syllabification. This sort of graded categorization judgement might be akin to what is referred to in the literature as ambisyllabicity, where a single segment serves both as the coda of one syllable and onset to the next (Kahn 1976).

As for the strongly ungrammatical examples, the model was unable to reproduce the input on the output layer. Instead, it produced near random output, which results in a high error rate. If error is taken as a measure of grammaticality, the response of the model is reasonable. One can contrast this with the cases described above, where prototypical items were reproduced with little error.

Perhaps the most illuminating was the response of the model to the third class of test items. In all three of these cases, the network edited the illicit forms and produced acceptable syllable structures. Given the input *VCCS, the model added a V to the end, and resyllabified to give the good form VC.CSV. In the second case, the input string ?CV:C was modified to CV:CV. This is interesting because long vowels in closed syllables appear to be only marginally acceptable at normal rates of speech in Turkish (Sezer 1986). This, then, is a reasonable response to this input string. The final form was the most interesting for us, since it corresponded to an illegal surface form which is assumed to be an existing underlying form in Turkish. This was the input string *CVCS. The model responded with the output CV.CVS. Note that the vowel here was not simply added to the end of the string, but inserted between the second C and the S. This is noteworthy because this is precisely the environment in which Turkish epenthesizes vowels. (See example (1).)

We find these results interesting for a number of reasons. In the first place, the model was able to reproduce well-formed strings correctly, and unable to reproduce those that were ill-formed. The model's response to the generalization test demonstrated that it had not only induced generalizations about canonical syllable structure, but that these generalizations had processing consequences, and were used to constrain novel outputs. Finally, these constraints were induced through exposure to only positive data.

The results given above are encouraging. However, the auto-associator is an example of a very basic type of network, and as such displays certain limitations. For example, the encoding of inputs and outputs as static, fixed-length patterns is an unrealistic representation of language. Language is clearly a

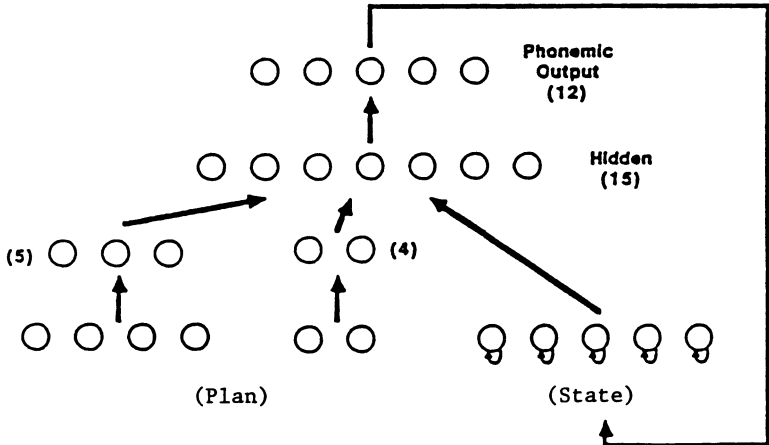
sequential phenomenon, and ideally this should play a role in the representations in language models. Furthermore, the representations used in this model presuppose certain theoretical constructs. The input to the model overtly classifies consonants as onsets and codas, while the form of the test set explicitly assumes the reality of underlying representations. The objection might be raised that the success of this simulation depends on such questionable features of the model.

For this reason we ran a second simulation, using a more sophisticated architecture which avoids many of the limitations of the first model.

IV. Simulation II

The architecture used in this simulation is based on work by Michael Jordan (1986b). In this model the output data represents words, rather than simply CV skeletons. However, these words are produced over time instead of as static output patterns. In the model, output at any given time represents only one phoneme; a word is represented as a string of phonemes on successive output cycles. The model used the architecture diagramed below.

(9)



(Jordan 1986)

Input to the model is divided in two parts. The first input bank, referred to as the "plan", contains no phonological information. Instead, this plan is an arbitrary vector that can be taken to represent the concept which a given phonological string expresses. The mapping between the input and output patterns is the mapping from this abstract "concept" to a phonological output. A concrete example will help clarify the input-output relationship. Suppose that the desired output is the phonological string

cat. The input (the "plan") is only an arbitrary vector of numbers, for example [11011]. This plan is presented and held constant as the network produces, in sequence, the three outputs corresponding to /c/, /a/, and /t/. Although there is no phonological information present in the plan, the network is trained to respond to the cue [11011] by producing the output sequence c-a-t. This training is effected in the same way as in the earlier model.

However, there is a difference between this model and the one described earlier. Notice in the diagram that the input consists not only of the "plan", but also of a bank of units referred to as the "state". As the network produces a response at time t1, that response is fed back onto the "state" units, and forms a part of the input at time t2. This means that as the network learns, it is given two pieces of information: (i) a plan, which triggers the production of a particular sequence, and (ii) the current state, which serves as a context, telling the network what part of that sequence was produced last. These two inputs in combination aid the network in learning what step of the sequence is the next to be produced.

The intent of this second simulation was to examine whether the positive results of the first model could be reproduced given the constraints imposed by a sequential network. This is a desirable outcome, since this architecture has certain advantages over the auto-associator used in Simulation I. It avoids the difficulty of a fixed-length representation of variable-length patterns, since the length of the output pattern is no longer frozen into the architecture. In addition, the model is able to examine data involving phonological alternations without having to take a stand on the psychological reality of underlying representations. A form which never appears on the surface need not be represented as input.

In this simulation, the training set consisted of fourteen roots and two morphological variants of each, for a total of 28 possible forms.

The input was given as described above, and the output was processed dynamically. That is, the phonemes of each word were represented sequentially, one per cycle, over the eleven bits of the output layer. These eleven bits encoded syllable location (onset, peak, and coda) and a ten-bit modified distinctive feature matrix, shown below.

- (10) 1. syllable placement
2. vocalic
3. consonantal
4. front/back
5. voiced
6. nasal
7. high
8. low
9. stop
10. strident

11. round

The network was trained on a subset of 24 of the 28 possible forms. These involved the 14 different lexical items, with one or both morphological variants of each.

V. Results

As stated above, the task of the current model was to take in an arbitrary code for a word and produce the correct surface form. The simulation ran for 10000 epochs, with training as described in Section III. The network was then tested on a set of four novel patterns. Testing involved giving an input pattern corresponding to a stem plan which the network had seen, combined with a novel morpheme plan. For example, if the training set had included a given word only in the nominative, the test set asked for the accusative form. If the network had seen only the accusative form of the word during training, it was tested on the nominative.

The task facing the network was to produce the phonological form of a word, given an arbitrary plan corresponding to that word. However, the hypotheses being tested were the same as in the original simulation. The model develops a set of connection weights in the process of learning to produce the correct output patterns. These weights permit the network to output correctly the forms it has learned, and the generalizations encoded in these weights are the equivalent of "well-formedness conditions" which impose themselves on the outputs. As the network develops a set of weights which allow it to produce the correct phonological forms, those weights act as constraints on future outputs. The prediction here, as in the first simulation, is that those constraints will result in phonological alternations that correspond to real processes occurring in the language. The results of the test were very encouraging in that they confirmed this prediction.

The test set consisted of the plans corresponding to the four following output forms:

(11)	<i>bakir</i>	'copper' (nom)
	<i>Çiiri</i>	'era' (acc)
	<i>garipi</i>	'strange' (acc)
	<i>fikir</i>	'idea' (nom)

The first, *bakir*, was reproduced perfectly. This was the nominative, or unaffixed, form. The next two entries in the test set were in the accusative case, and these were output as vowel-final, as required. In one case (*Çiiri*) this additional vowel harmonized with those of the stem. Vowel harmony is a phonological process evident in Turkish. Once again the model extracted this generalization even though this was not an original intent of the simulation. The third entry, *garipi*, was produced with the final high vowel which marks the accusative case. However, in this case the

vowel did not harmonize with the stem vowel, so that the actual output was closer to an /u/ than to an /i/.

Fikir, the last entry in the test set, is the most interesting. The network was trained on the accusative form of this word, which is *fikri*. Notice that if the network creates a nominative form simply by eliminating the accusative affix, the expected output is **fikir*. As was shown in Section II, *fikir* is not an acceptable phonological form in Turkish. Although this model was given no information on syllable-structure constraints, it correctly "epenthesized" a high vowel between the stop and the sonorant.

In summary, the results of both simulations strongly suggest that syllable well-formedness information can be induced from examples of positive data, and this information is then used by the network, pressuring outputs to conform to canonical patterns. These results have certain implications. First, constraints on syllabic well-formedness need not be taken as given, but can be learned (See McCarthy and Prince 1986 for a similar point). In addition, the constraints which are learned are not discrete statements that function independently of the data. Instead, these are soft constraints, intimately connected with the data and subject to continual modification. While some have argued that this context-dependency is an inherent problem in connectionist language models, we suggest that this is in fact a desirable outcome. It is the fact that these constraints are not independent which allows them to account for data that might not fall neatly into discrete classes.

BIBLIOGRAPHY

- Chomsky, Noam (1980). "Rules and Representations." *Behavioral and Brain Sciences* 3, 1-61.
- Clements, G.N. and S.J. Keyser (1983). *CV Phonology: a Generative Theory of the Syllable*; Linguistic Inquiry Monograph #9. Cambridge, Mass.: MIT Press.
- Clements, G.N. and E. Sezer (1982). "Vowel and Consonant Disharmony in Turkish" in H. van der Hulst and N. Smith (eds): *The Structure of Phonological Representations Part II*. Cinnaminson, New Jersey: Foris Publications.
- Hayes, B. (1989). "Compensatory Lengthening in Moraic Phonology." *Linguistic Inquiry* 20, #2.
- Jordan, M. (1986a). "Introduction to Linear Algebra in Parallel Distributed Processing" in Rumelhart, D.E., J.L. McClelland, and the PDP Research Group (1986).
- Jordan, M. (1986b). *Serial Order: A Parallel Distributed Processing Approach*. ICS Report #8604. UC San Diego.
- Kahn, D. (1976). *Syllable-based Generalizations in English Phonology*. Bloomington, Indiana: Indiana University Linguistics Club.
- Kornfilt, J. (1986). "Stem Penultimate Empty C's, Compensatory Lengthening and Vowel Epenthesis in Turkish" in L. Wetzels and E. Sezer (eds): *Studies in Compensatory Lengthening*. Riverton, New Jersey: Foris Publications.
- McCarthy, A. and S. Prince (1986). "Prosodic Morphology." ms.
- McClelland, J.L., Rumelhart, D.E., and the PDP Research Group. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 2: Psychological and Biological Models*. Cambridge, Mass.: MIT Press/Bradford Books.
- Pinker, S. and A. Prince (1987). "On Language and Connectionism: An Analysis of a Parallel Distributed Processing Model of Language Acquisition." *Cognition*, 28, 73-193.
- Rumelhart, D.E., G. Hinton, and R.J. Williams (1986). "Learning Internal Representations by Error Propagation" in Rumelhart, D.E., J.L. McClelland, and the PDP Research Group (1986).
- Rumelhart, D.E., J.L. McClelland, and the PDP Research Group (1986a). *Parallel Distributed Processing: Explorations in the*

Microstructure of Cognition. Volume 1: Foundations. Cambridge, Mass.: MIT Press/Bradford Books.

Schane, S. (1968). *French Phonology and Morphology.* Cambridge, Mass.: MIT Press.

Sezer, E.F. (1986). "An Autosegmental Analysis of Compensatory Lengthening in Turkish" in L. Wetzels and E. Sezer (eds): *Studies in Compensatory Lengthening.* Riverton, New Jersey: Foris Publications.

Smolensky, P. (1988). "The Proper Treatment of Connectionism." *Behavioral and Brain Sciences*, 11, 1-74.