Recurrent Neural-Network Learning of Phonological Regularities in Turkish

Jennifer Rodd
Centre for Cognitive Science
University of Edinburgh
2 Buccleugh Place
Edinburgh, EH8 9LW, UK
jenni@cogsci.ed.ac.uk

Abstract

Simple recurrent networks were trained with sequences of phonemes from a corpus of Turkish words. The network's task was to predict the next phoneme. The aim of the study was to look at the representations developed within the hidden layer of the network in order to investigate the extent to which such networks can learn phonological regularities from such input. It was found that in the different networks, hidden units came to correspond to detectors for natural phonological classes such as vowels, consonants, voiced stops, and front and back vowels. The initial state of the networks contained no information of this type, nor were these classes explicit in the input. The networks were also able to encode information about the temporal distribution of these classes.

1 Network Architecture

The network used is a simple recurrent network of the type first investigated by Elman (Elman, 1990). It consists of a feedforward network, supplemented with recurrent connections from the hidden layer. It was trained by the back-propagation learning algorithm (Rumelhart, Hinton and Williams, 1986). The ability of such networks to extract phonological structure is well established. For example, Gasser (Gasser, 1992) showed that a similar network could learn distributed representations for syllables when trained on words of an artificial language. Figure 1 shows the architecture of the network. Within this network architecture, four different network configurations were investigated. These all had 28 units in both the input and output layers; they varied only in the number of units in the hidden layer, ranging from two to five.

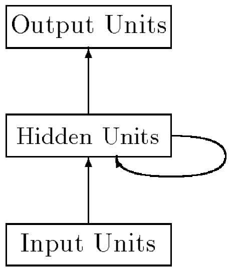

Figure 1: Network Architecture

All connections in the network have an inherent time delay of one time step. This has the result that the recurrent connections between units in the hidden layer give the network access to a copy of its hidden-layer activations at the previous time step. The delay also has the effect that it takes two time steps for any information to propagate from the input layer through to the output layer. The network is fully connected

The input to the network is a series of sequentially presented phonemes from a corpus of 602 Turkish words. Each phoneme is represented by a 28-bit vector in which each of the 28 Turkish phonemes present in the corpus is represented by a different bit. Whenever a particular phoneme is present, the corresponding bit is flipped on. This sparse encoding scheme was taken from Elman (Elman, 1990), and ensures that each vector is orthogonal to every other vector. Thus the network is given no information about the similarity between different phonemes.

Each word is presented to the network as a series of such phonemes, with each phoneme presented in a successive time step. Each word constitutes a discrete training item, i.e. the network is not required to segment words. During training, the weights are updated after each training item, i.e. after every word.

The task of the network for each input phoneme

Jennifer Rodd (1997) **Recurrent Neural-Network Learning of Phonological Regularities in Turkish.** In T.M. Ellison (ed.) *CoNLL97: Computational Natural Language Learning,* ACL pp 97-106.

is to predict the following phoneme. Due to the two time delays present in the structure of the network, this prediction task is constructed by requiring the units in the output layer to show the pattern of activation present at the input layer on the preceding time step. The network's structure ensures that none of this information from the preceding time step could have propagated through to the output units by this time, and so the task is a genuine prediction task.

The networks are trained using the Xerion simulator, using the back-propagation learning algorithm. A momentum term is used to reduce the training time.

2 Vowel Harmony in Turkish

The networks are trained on Turkish words. Turkish was chosen for its well-documented and interesting phonological structure; in particular its vowel harmony (Lewis, 1967). It has already been shown by Hare (Hare, 1990) that vowel harmony can be modelled successfully by using connectionist models. She has used recurrent networks of the type developed by Jordan (Jordan, 1986) to model vowel harmony in Hungarian. This model successfully accounts for many of the complexities of Hungarian vowel harmony, and predicts the behaviour of both harmonic and transparent vowels.

Unlike Hare, however, I am not concerned with the modelling of a particular phonological process. I am interested in investigating what information such a network can learn about the structure of a language, given minimal information. Hare's networks are given a featural description of the phonemes involved, and so have an inherent measure of their similarity, and therefore of the phonological classes. Further, Hare's networks are given only sequences of vowels. My networks are given both the vowels and the intervening consonants, and therefore have the possibility to simultaneously learn a wide range of phonological regularities. Despite the presence of intervening consonants, it was expected that such networks could learn the basics of vowel harmony. Therefore, before discussing the networks in detail, let me first outline vowel harmony in Turkish.

Clements and Sezer (Clements and Sezer, 1982) describe vowel harmony as a "system of phonological organization according to which all vowels are drawn from one or the other of two (possibly overlapping) sets within harmonic spans in the word". Turkish is an example of what Clements and Sezer call a "symmetrical vowel harmony system". Words consist of a stem and a sequence of suffixes. The vowels in the stem do not alternate, while the vowels in the suffixes alternate such that they agree with the nearest non-alternating vowel. Specifically, in Turkish, each word will typically only contain vowels with the same value for the feature [±front]. The fronting of the stem vowels determines the fronting of the suffix vowel(s). The fronting of the vowels within the stem itself is usually uniform. There is also vowel harmony for the feature [±round]. Any high vowel in the second syllable of the word (or later) has the same value for the feature [±round] as the vowel in the preceding syllable. Low vowels after the first syllable are all [−round].

Table 1: Phonological features for Turkish vowels

	a	e	ı	i	o	ö	u	ü
[±front]	−	+	−	+	−	+	−	+
[±round]	−	−	−	−	+	+	+	+

Turkish also displays consonant harmony. The consonants /k, g, l/ each have two phonetic shapes, which differ in the value of the feature [±front]. The value for this feature is determined by the fronting of the vowels in the word. However, this phenomenon of consonant harmony can clearly not be considered in this study, as the two allophones for these consonants are represented by the same phoneme in the input data.

Clements and Sezer (Clements and Sezer, 1982) describe in detail a number of exceptions to these basic harmony rules, and provide an account for these irregularities in terms of the presence of opaque vowels and consonants in the underlying representation of the segments. Exceptional cases include the existence of some disharmonic polysyllabic roots. Disharmonic suffixes also exist, in which at least one vowel fails to alternate under any circumstances.

The corpus used for this study contained 601 Turkish words. 91% of these words showed harmony for the feature [±front]. The other 9% contained both front and back vowels.

3 The Networks

I will now discuss the results of four simulations using networks of the type described above. The only difference between the architectures of the networks is the number of units in the hidden layer; the training data remained the same for all simulations. I will discuss in detail results from individual training runs. It was found that different runs starting with different initial randomized weights produced results that were remarkably consistent. Therefore, for clarity, I will discuss only one set of results for each simulation. The networks differed in the number of times the corpus was seen in training; this ranged from 77 for Network 4, to 91 for Network 1. These numbers were determined by the point at which the network reached a particular tolerance level for its error score. For each network, a value for the tolerance in the error was chosen that consistently enabled the network to settle on the solutions described. Lowering this tolerance resulted in a failure of the learning algorithm to converge, while increasing the tolerance resulted in the network learning very specific regularities about the training set. In such cases the regularities learnt depended on the initial weights. The tolerance levels were therefore chosen to produce networks that consistently learned general solutions to the prediction task.

Network 1

For this network, the hidden layer has only two units. The results of training such a network on the corpus are extremely clear and consistent. The training algorithm for the particular network I shall consider here converged after 54585 training examples (i.e., each training example was seen approximately 91 times). The restriction to only two hidden units allows the network to encode a single regularity in the structure of the input. The strongest phonological regularity present in Turkish, as with most languages, is the alternation of vowels with consonants. The corpus contains eight vowel types, but very few vowel clusters. Thus, when the network has seen a vowel, it can be almost certain that the following phoneme will be a consonant. Similarly, although consonant clusters are present in the corpus, single consonants are more frequent. Therefore directly after a consonant, a vowel is the best prediction.

Consistent with this hypothesis, analysis of the network after training shows that indeed one of the hidden units has learned to respond most strongly to vowels, and the other to consonants. This can be seen by looking at the hidden-unit activation levels one time step after presentation of a single phoneme. The activation of units in the network was always positive, with an activation level of 1 corresponding to the maximum activation of a unit. These activation levels are shown in Figures 2 and 3, which clearly demonstrate that the two hidden units have been used by the network to classify each of the input patterns as either a vowel or a consonant.

Also of interest are the weights of the connections between the two hidden units and the output layer. Intuitively, we would expect that Hidden Unit 0, which responds most strongly to consonants, would have the strongest connection to those output units

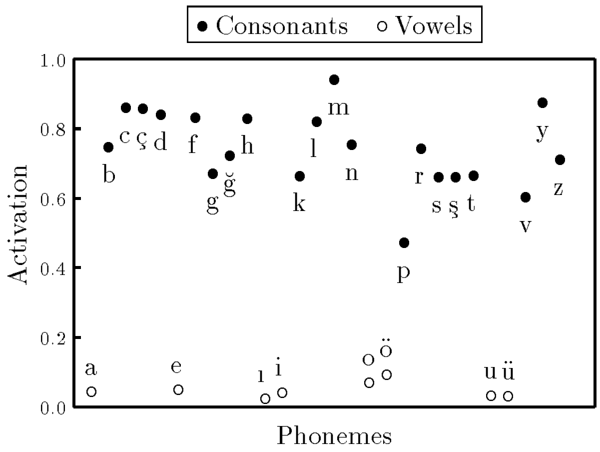

Figure 2: Activation of Hidden Unit 0, Network 1 in response to single phonemes at the input layer

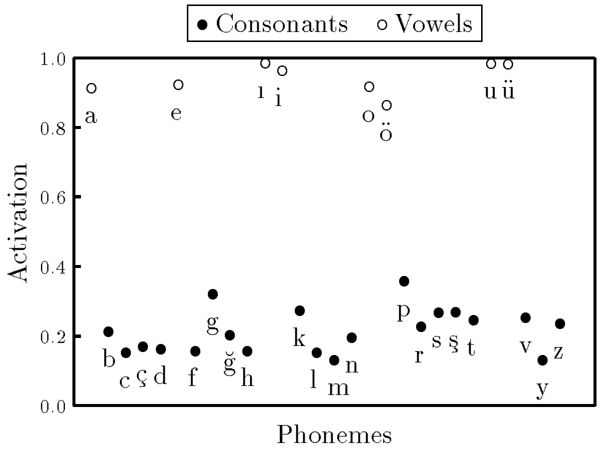

Figure 3: Activation of Hidden Unit 1, Network 1 in response to single phonemes at the input layer

that represent vowels. This would encode the fact that when the network has just seen a consonant, it should predict a vowel. Conversely, we would expect Hidden Unit 1 to be most strongly connected to consonants. Indeed, this general pattern of connectivity is found. However, the large variance in the frequencies of the various phonemes makes it hard to make direct comparisons between the values of the connection weights. In other words, Hidden Unit 1 may be more strongly connected to some high-frequency vowels than to some of the consonants such as /h/, which has only 23 tokens in the corpus of 4198 phoneme tokens.

A clearer pattern emerges by looking instead at the activation levels of the output units, two time steps after the network was presented with particular phonemes. This two-time-step delay is simply to account for the two time steps necessary for the information to propagate through the network. This is equivalent to asking the network what phoneme it expects to follow the single phoneme that has been presented.

These activation levels were frequency adjusted by dividing the activation levels for the units representing each phoneme by the frequency of that phoneme in the corpus. This adjustment compensates for the networks' tendency to predict more frequent phonemes, and allows us to observe any other trends superimposed on this frequency effect. In fact, rather than absolute frequency, a proportional frequency measure is used.

Figures 4 and 5 show the frequency adjusted activation of the various units in the output layer after the network has been presented with /ı/ and /d/ (the frequency adjustment makes the units of activation for these graphs arbitrary). This vowel-consonant pair was chosen because they have similar frequencies in the corpus (176 and 177 out of a total of 4198 respectively). The output-layer activation levels for the other 25 phonemes show the same pattern, with consonants activating units representing vowels, and vice versa.

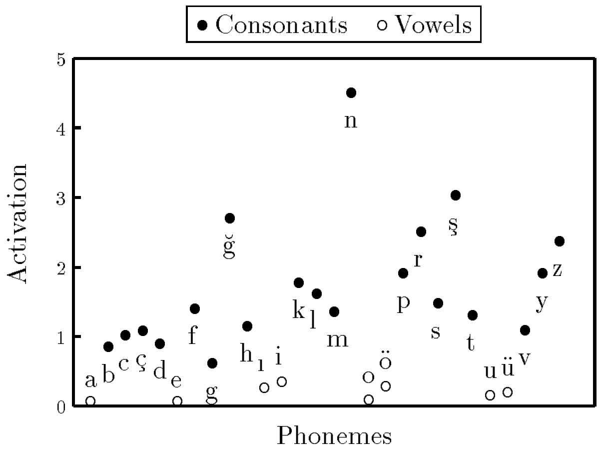

Figure 4: Frequency adjusted activation of units at the output layer in response to "/ı/" in the input layer

Thus we have seen that, given only two hidden units, the recurrent network learns the difference between the distributional properties of vowels and consonants. It has divided the group of input phonemes into two natural classes, and it uses these representations to predict the appropriate phoneme in the output layer. It is of interest that the consonant that is closest to the vowels in terms of activation level is not one of the liquids, such as /y/ or /l/,

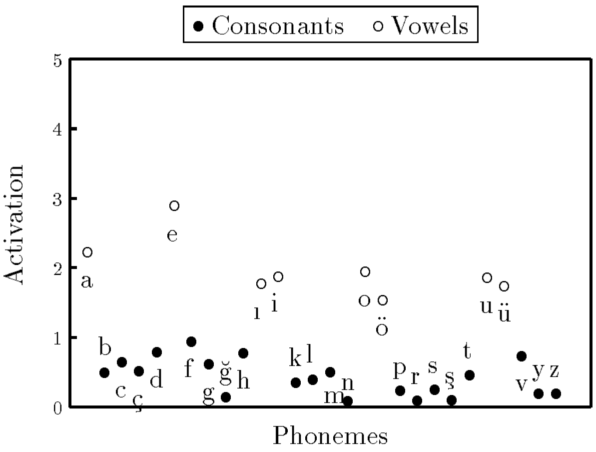

Figure 5: Frequency adjusted activation of units at the output layer in response to "/d/" in the input layer

which are featurally most similar to vowels. It is the stop consonant /p/. This underlines the fact that the network is making the division on purely distributional grounds. The fact that /p/ is treated as the most "vowel-like" of the consonants stems from the fact that it is the consonant that occurs as the first consonant of a consonant cluster in the highest proportion of its instances. This can be seen in Table 2, which gives the total frequencies for the different consonants, as well as the number of times they participate in the first and second positions within a consonant cluster; the total count includes consonants which participated in clusters, and consonants which appeared on their own between vowels. /p/ occurs only 55 times in the corpus, and in 32 of these it is followed by another consonant (/l/, /r/ or /t/). In this respect, its distribution is more vowel-like then any other consonant.

Table 2: Total frequencies for consonants in the corpus, and the number of times they appear in consonant cluster initial (CI) and consonant cluster final (CF) positions

	Total	CI	CF
b	94	0	3
c	33	1	15
ç	58	6	8
d	177	2	77
f	18	3	3
g	55	0	6
ğ	59	9	0
h	22	5	0
k	223	47	25
l	311	52	146

	Total	CI	CF
m	167	28	72
n	313	82	1
p	55	32	1
r	298	65	27
s	88	16	10
ş	90	41	2
t	167	26	49
v	23	12	3
y	112	17	6
z	46	11	0

Network 2

This network differs from Network 1 only in that it has three hidden units. This network converged after training on 47963 examples. It was expected that not only would it learn the vowel-consonant distinction, but it should be able to use the additional hidden unit to encode another phonological regularity found in the corpus. It was thought that the extra hidden unit might enable the network to learn basic vowel harmony, but this is not the case.

As anticipated, the network learns the vowel-consonant distinction in an identical way to Network 1. Hidden Unit 0 and Hidden Unit 2 in this simulation behave almost identically to the two hidden units in Network 1. Hidden Unit 0 responds maximally to vowels, while Hidden Unit 2 responds maximally to consonants. The graphs of their activations in response to single-letter inputs are extremely similar to Figures 2 and 3.

This leaves the question of what Hidden Unit 1 is being used for. Figure 6 shows the activation of Hidden Unit 1 in response to the presentation of single phonemes to the input layer. This shows that it is clearly not involved in the consonant-vowel difference; for vowels it is difficult to see any pattern in what it is learning, and it is certainly not learning vowel harmony. I have already suggested that there are differences between the consonants in terms of their participation in consonant clusters, and it is these differences that this unit seems to be capturing.

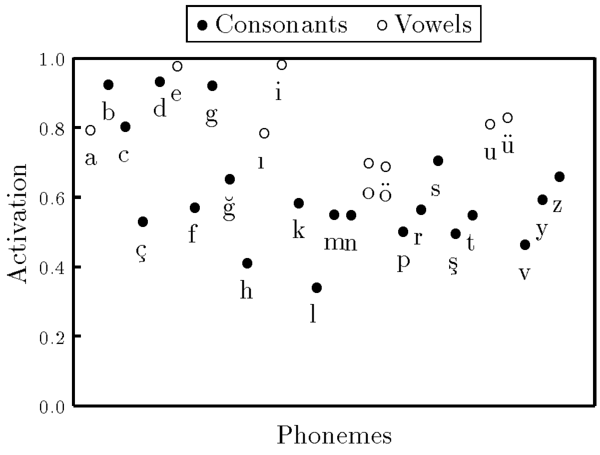

Figure 6: Activation of Hidden Unit 1, Network 2 in response to single phonemes at the input layer

In Turkish, voiced stop consonants are rarely followed by consonants. In the corpus, /b/ and /g/ are never followed by another consonant, while /d/ is only twice followed by a consonant. The consonant /c/ also only occurred once in such a cluster, although it is worth noting that its overall frequency in the corpus (33 out of a total of 4198 tokens in the corpus) is lower than those of the three voiced stop consonants (see Table 2). Thus, when the network sees one of these consonants, is can be confident in its prediction of a vowel as the following phoneme. Indeed if we look at Figure 6, we can see that for the consonants there is a cluster of high activation for the voiced stop consonants, /b/, /d/ and /g/, while /c/ has a slightly lower activation.

This suggests that Hidden Unit 1 is involved in encoding the fact that some consonants are more likely to be followed by consonants than others, i.e. it is learning sonority. "Sonority" is the characteristic that is involved in determining what segments may legitimately appear adjacent in clusters. If the role of this unit is to make predictions about consonant clusters, we would expect its activity to have the effect of turning off the output units corresponding to consonants. This is indeed the case. The connection weights between Hidden Unit 1 and the output units are all nearly all strongly negative. The exceptions to this are the output units representing the consonants /l/, /m/ and /n/, which have small negative, or in the case of /n/ positive, connections. The activation of /l/ and /m/ reflects the fact that these consonants are likely to occur in the final position of a consonant cluster, while /n/'s activation is probably simply due to its high frequency. It is the most frequent consonant in the corpus, with 313 tokens (out of a total of 4198 tokens).

Also of interest are the weights of the recurrent connections within the hidden layer. Hidden Unit 1 receives inhibition from Hidden Unit 2 via a strong negative connection. The connection from Hidden Unit 0 is small but positive. This means that Hidden Unit 1 will be maximally active when the previous phoneme to have an influence in the hidden layer was a vowel. This is consistent with the idea that the unit is responding to the presence of a consonant that was preceded by a vowel, i.e. a consonant that may be the start of a consonant cluster.

This means that the fact that this unit is also strongly activated for the vowels is not a problem. Vowels are almost always preceded by a consonant. Therefore, Hidden Unit 1 will be inhibited by the activation of Hidden Unit 2. Thus, activation of Hidden Unit 1 by a vowel in the input layer will be insufficient to cause it to inhibit the prediction of a consonant as the following phoneme.

To summarize, Hidden Unit 1 is allowing for the fact that in some instances a consonant can follow another consonant. In general, it acts to reduce the activation of post-consonant consonants, but this inhibition is less in the cases where the initial consonant is not a voiced stop. There is also less inhibition of those consonants that are more frequently found at the ends of consonant clusters, than of any of the other consonants.

Network 3

This network was produced by adding a further unit to the hidden layer. Training of this network converged after 46759 training examples. Let us now look at the behaviour of these four hidden units in turn.

The behaviour of Hidden Unit 2 is probably the simplest to explain. It is simply a consonant detector such as those we have seen before. Accordingly, it inhibits the activation of the units representing consonants in the output layer, while strongly activating those units representing the more frequent vowels. Again, /p/ is treated as the most vowel-like of the consonants.

Figure 7 shows that Hidden Unit 3 has divided the input space into three categories. It is most strongly activated for the vowels /a/, /ı/, /o/ and /u/, namely the [−front] vowels. It also responds to the [+front] vowels, but the level of response is lower for these vowels. Lower again is the unit's response to the consonants. The activation of this unit has two main effects. Firstly, the unit has a strong negative connection to Hidden Unit 1. We will return to the effect of this later.

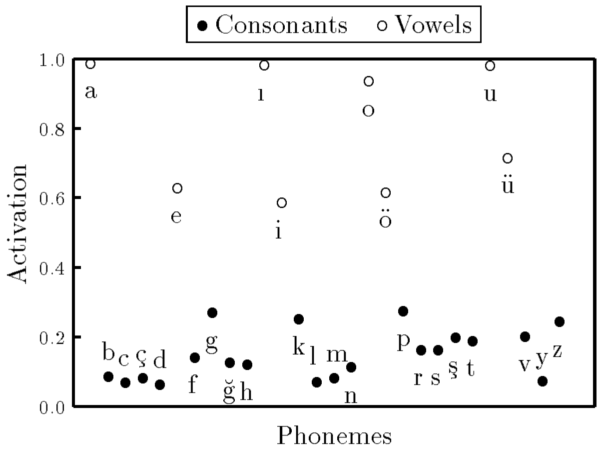

Figure 7: Activation of Hidden Unit 3, Network 3 in response to single phonemes at the input layer

The more immediate effect of this unit on the output layer is similar to that of the vowel detectors already seen. It acts to reduce the activation of the output units corresponding to vowels. This prevents the network from predicting a vowel immediately after another vowel. The unit's connections to consonant units in the output layer are less strongly negative, or in the case of /k/, /l/ and /p/, positive. These consonants do appear to follow back vowels in a disproportionate number of cases.

Now we come to the third hidden unit, Hidden Unit 1. This is possibly the most interesting. Its response to input, shown in Figure 8, shows no clear pattern. Note also that no phoneme raises its activation above 0.6. It responds more strongly to the consonants, except for /h/, /p/, /v/ and /z/. What makes these consonants different is that they are disproportionately likely to begin consonant clusters (see Table 2). Thus, this unit is active for consonants that are most likely to be followed by a vowel.

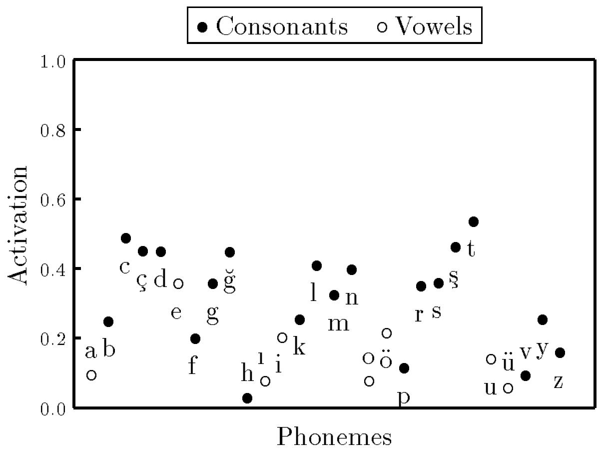

Figure 8: Activation of Hidden Unit 1, Network 3 in response to single phonemes at the input layer

Earlier, I mentioned that Hidden Unit 3 has an effect within the hidden layer. The recurrent connection within the hidden layer with the second largest weight is the connection from Hidden Unit 3 to Hidden Unit 1, which is large and negative. Thus Hidden Unit 1 is turned off when the preceding input was a [−front] vowel. Hidden Unit 1 is also turned off by Hidden Unit 2, which, as we saw earlier, is activated by consonants. So we have a unit whose response is greatest for a [+front] vowel on the preceding time step, followed by a consonant that is unlikely to be starting a consonant cluster. Hidden Unit 1 also has a positive self-recurrent connection, so that once it has been activated it will remain active unless inhibited.

The rules of Turkish vowel harmony suggest that the phoneme most likely to follow a sequence of a [+front] vowel and a consonant, is another [+front] vowel. Therefore, Hidden Unit 1 should activate

[+front] vowels in the output layer. The weights from Hidden Unit 1 to the output units representing vowels are given in Table 3. This suggests that rather than activating [+front] vowels, its action is instead to reduce the activation of the [−front] vowel, in particular, /a/, /ı/ and /u/, which are the most frequent of the [−front] vowels.

Table 3: Weights to output layer units representing different vowels from Hidden Unit 1

Vowel	a	e	ı	i	o	ö	u	ü
±front	−	+	−	+	−	+	−	+
Weight	-7.1	+1.2	-7.2	-0.4	+1.3	-0.6	-3.9	-0.6

This asymmetry between the network's treatment of front and back vowels has implications for its performance. One measure of the network's performance is to input a single vowel and to look at its predictions in the output layer. Rather than looking at the output in the time step when the network is predicting the phoneme to follow the vowel in question, I have looked at the output in the following step. This is the time step when the network is required to predict the phoneme two time steps on from the vowel, which is more likely to be a vowel. The vowels predicted most strongly should agree in fronting with the input vowel.

Looking at such output units shows that, although the network shows a preference for vowels of the same [±front] value, there is an asymmetry in performance. The difference between the output in the units representing front and back vowels is approximately twice as large when the input is a back vowel. In other words, the fact that a single unit is used to encode whether an input is [+front] or [−front] has meant that the network has in effect learnt fronting harmony better for back vowels than for front vowels. Looking at the training corpus reveals that of the 544 harmonic words in the corpus, 50.6% contain only back vowels, while the remaining 49.4%contain only front vowels. Of the 57 disharmonic words in the corpus, 53% had a front vowel as the first vowel in the word. These small differences provide a possible explanation for the fact that it is the back vowels for which vowel harmony is better learned, but it is insufficient to explain the large asymmetry in the network's performance. This difference must therefore be seen as a result of the limitations of the network architecture, and not a direct result of the data it was trained on.

However, despite this, Hidden Units 1 and 3 together have enabled the network to learn fronting harmony.

This leaves us with just one hidden unit to account for, Hidden Unit 0. Its pattern of activation in response to inputs of individual letters shows no obvious categorization. It responds highly to vowels, as well as to most of the consonants, especially /h/, /m/, /l/, /v/ and /y/. It is difficult to see that this unit is contributing anything of importance to the straightforward mapping from input to output. Thus, the key to its behaviour must lie within the hidden layer.

Firstly it has a strong, negative self-recurrent link. Thus, once the unit is activated, it will, if left to itself, continually turn itself on and off at successive time steps. The strongest connection within the hidden layer is the positively weighted connection from Hidden Unit 2 to Hidden Unit 0. Thus, this unit is on when the preceding input phoneme is a consonant. It also has a positive link forward to Hidden Unit 2. This will result in Hidden Unit 2 being activated by a consonant, and then being reactivated two time steps later. Thus, Hidden unit 2 appears to oscillate in opposition to Hidden Unit 0. The behaviour of this unit is clearly complex. It is encoding something about the temporal structure of the input, rather that making direct predictions on the basis of the last input phoneme. The exact details of this behaviour are beyond the scope of this paper. However, one result of its behaviour is worth mentioning.

Consider the activation of Hidden Unit 0 in response to the input of a single vowel phoneme at the input layer. In the first time step it responds with activations ranging from 0.45 for /ö/ to 0.92 for /e/. Its activation then drops on the following time step in proportion to its initial activation, i.e. the negative self-recurrent link acts to reduce its activation most in those cases where it is most active. Then, on the following time step, its activation increases again for all the vowels. Activation levels range from 0.80 for /ö/ to 0.96 for /e/. It is probable that Hidden Unit 0's oscillatory behaviour is allowing the network to capture useful information about the vowel-consonant alternation over time.

To summarize, this network has only four hidden units, and yet it shows complex behaviour. It has encoded much information about vowel-consonant alternation and vowel harmony. Looking at the output layer also shows that it has some knowledge about consonant clusters. For example, comparing the consonants /d/ and /n/, /n/ is a more frequent phoneme, with a total of 313 tokens in the corpus to /d/'s 177. However, /d/ appears second in a consonant cluster 77 times, while /n/ appears in this position only once. If we look at the activation of the output units representing these phonemes, we see that, after consonants frequently in the cluster initial position, such as /n/ itself or /r/, /d/'s activation is over 20 times greater than that of /n/. Clearly, this network has also learned about which consonants are likely to fall in particular positions in consonant clusters.

What is also clear, however, is that as the networks become more complicated they become increasingly harder to analyse. No longer do we have only simple detectors for phonological natural classes such as consonant and vowels; i.e the network is able to use the recurrent links to encode complex temporal properties of the input. We also see that the network shows behaviours that are difficult to attribute to individual hidden units.

Network 4

This network has 5 hidden units, and saw 46157 training examples. The hidden units show many of the characteristics already discussed, in terms of learning about the properties of consonant clusters. Most of the network's behaviours are extremely complex, and not sufficiently different from patterns already seen to make them of significant interest. Of more interest is the ability of this network to capture vowel harmony, and it is to the units responsible for this that I will limit my discussion.

Hidden Unit 4 is used as a straightforward vowel detector such as we have seen before. It is activated most strongly by the input units representing the 8 vowels. Its connections to the output units representing vowels have high negative weights, to prevent the prediction of a vowel after the network has seen a vowel. Its self-recurrent connection also has a large negative weight; vowel sequences were very rare in the corpus.

Hidden Units 0 and 2 are involved in the network's learning of vowel harmony. They both respond to consonants as well as vowels, but for the moment let us consider just their responses to the activation of the input units representing vowels. Hidden Unit 2 responds strongly to the [−front] vowels /a/, /ı/, /o/ and /u/, but shows negligible activation in response to the [+ front] vowels /e/, /i/, /ö/ and /ü/. Hidden Unit 0 shows the reverse pattern, except that its response to the [+front] vowel /ö/ is not as large as that to the other [+front] vowels. The most likely explanation for this is that it is due simply to the low frequency of this vowel in the corpus. It is the lowest-frequency vowel, with only 44 tokens. These patterns are shown in Figures 9 and 10.

Let us now look at the weights of the connections from these two units to the output layer. To show vowel harmony, we would expect to see the two

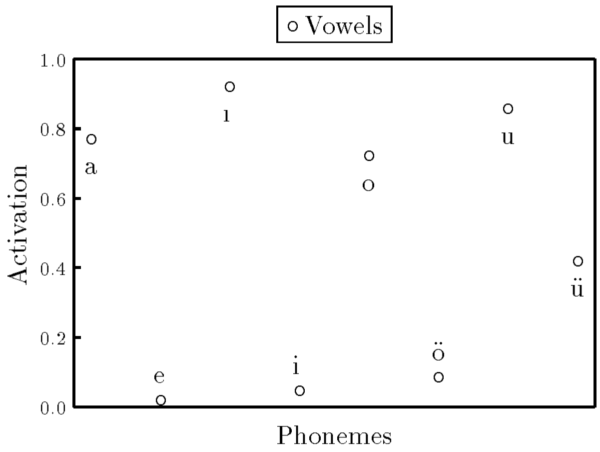

Figure 9: Hidden Unit 2, Network 4 activation in response to single phonemes at the input layer

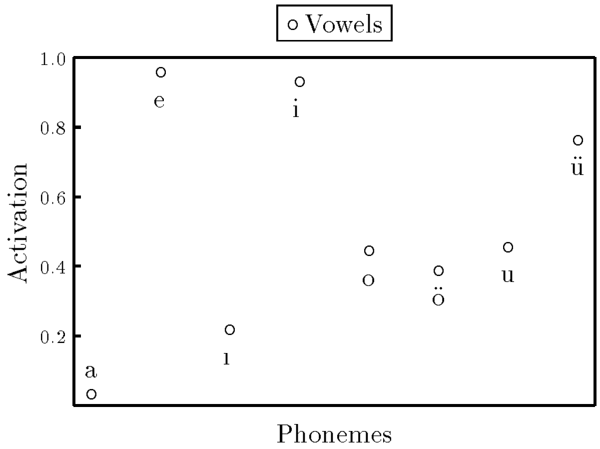

Figure 10: Hidden Unit 0, Network 4 activation in response to single phonemes at the input layer

hidden units activating the output units that represent vowels of the same value of [±front] as that to which they themselves are responding. The only output units to which Hidden Unit 2 has a positively weighted connection, are those representing the phonemes /a/ and /ı/, i.e. the most frequent [−front] vowels. The connections to the other two less frequent [−front] vowels are small and negative, and are smaller than the negative weights for the connection to the output units representing [+front] vowels. Thus the unit is using the fact that it has recently seen one of the [−front] vowels to predict the presence of another [−front] vowel, in particular /a/ and /ı/. Hidden Unit 0 does not show such a distinct pattern; rather it acts to inhibit Hidden Unit 2, and so prevents the prediction of a [+front] vowel.

Also of interest is the activation of the units in the output layer when the network is presented with a single phoneme. When this phoneme is a vowel, there is an interesting change in the prediction pattern with time. Immediately after the vowel, the activation for all vowels is low. Thus the network, as before, knows that a consonant almost always follows a vowel, and this general inhibition of the activation of output units representing vowels overpowers any effects of the vowel harmony units. However on the following time step, the network shows a strong preference for vowels with the value for [±front], consistent with the previous vowel. For example, the two most frequent vowels in the corpus are /a/, a [−front] vowel, and /e/, a [+front] vowel. If we input one of these vowels and then look at the response of the output units corresponding to the next two most frequent vowels (/i/ a [+front] vowel and /ı/ a [−front] vowel), the pattern shown in Table 4 emerges. The activation of the output unit is clearly higher where it agrees in fronting with the input vowel. For the lower-frequency vowels, the pattern is less strong, but still shows the vowel harmony effects. This and the earlier asymmetries between the learning of vowel harmony for the vowels of different frequencies, cannot be explained in terms of a interaction with harmony for the feature [±round]; such harmony was not observed to be significantly learned by any of the networks in this study. Presumably additional hidden-layer resources are necessary for the learning of such detailed regularities in the corpus.

Thus, not only are the units in the hidden layer successfully encoding the front-back distinction for vowels, but this is being translated at the appropriate time into the activation of output units consistent with vowel harmony of the feature [±front].

Table 4: Frequency adjusted output unit activation of vowels /ı/ and /i/ as predictions two time steps after /a/ and /e/

Input Vowel	[±front]	Output Vowel	[±front]	Output Activation
/a/	−front	/ı/	−front	6.73
/e/	+front	/ı/	−front	0.07
/a/	−front	/i/	+front	0.01
/e/	+front	/i/	+front	3.29

This persistence of the knowledge of the fronting of the vowels in the current word is most easily explained by the fact that Hidden Unit 2 has a very strong positive self-recurrent connection; this enables it to retain its high activation across the intervening consonants. As previously discussed, Hidden Unit 0 affects vowel prediction via Hidden Unit 2, and so knowledge about the presence of [+front] vowels also persists over time.

Therefore, unlike Network 4, Network 5 has devoted two hidden units to learning the regularities involved in vowel harmony. These two units are acting as detectors for the phonological natural classes of front and back vowels.

4 Conclusions

The four networks demonstrate the ability of simple recurrent networks to capture the temporal structure in phonological input. With an appropriate number of hidden units, these hidden units can become detectors for phonological natural classes such as vowels, consonants, voiced stop consonants, and front and back vowels. The prediction of the next phoneme at the output layer is based on the presence or absence of such classes of phonemes. However, unlike standard phonological theories, the classes are graded. For example, although the networks clearly treated consonants differently from vowels, some consonants are treated as more "vowel-like" than others.

It is worth remembering that these categories are derived purely on distributional grounds. The network has no knowledge of the articulatory or acoustic features of the phonemes. This perhaps explains why phonemes such as /j/ and /w/ are traditionally classed as consonants, despite the fact that they share many acoustic or articulatory features with vowels. On distributional grounds, their classification as consonants is undisputed.

Another observation worth noting is the change in the functional roles of the hidden units as their number increases. For example in the case of consonant clusters, in Network 2, one of the three hidden units is devoted to capturing the regularities in these clusters. In the larger networks, however, while their performance clearly indicates they have learned these regularities, it is less clear which units are implicated, and it appears that the function has become distributed across the hidden units.

To conclude, this study demonstrates that simple recurrent networks can extract phonological regularities purely on the grounds of the distributional differences between different phonemes. The resulting representations in the hidden layer correspond to groups that are treated as natural classes in phonological theories. While I am not suggesting that humans perform anything like this prediction task, what is clear is that the extraction of some of the generalizations important for the learning of phonological rules can be achieved on purely distributional grounds. In other words, the process of learning more complex phonological rules may be facilitated by the extraction of basic phonological classes prior to the learning of these rules.

The paper also demonstrates that while connectionist models containing many hidden units can be successfully used to model certain phonological processes in detail, restricting the number of hidden units allows us to investigate how representations for some of the basic phonological categories can be learned.

References

George N. Clements and Engin Sezer. 1982. Vowel and consonant disharmony in Turkish. In H. van der Hulst and N. Smith, editors, *The Structure of Phonological Representations, Part II*, pages 213-255, Dordrecht: Foris.

Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science*, 14:179-211.

Michael Gasser. 1992. Learning distributed representations for syllables. In *Proceedings of the fourteenth Annual Conference of the Cognitive Science Society*, pages 396-401.

Mary Hare. 1990. The role of similarity in Hungarian vowel harmony: a connectionist account. *Connection Science*, 2:123-149.

Michael I. Jordan. 1986. Serial order: a parallel distributed processing approach. In *ICS Report No. 8604*, UC San Diego.

Geoffrey L. Lewis. 1967. *Turkish Grammar.* Oxford: Clarendon Press.

David E. Rumelhart, Geoffrey E Hinton, and Ronald J. Williams. 1990. Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing, Volume 1*, pages 318-364, MIT Press, Cambridge, MA.