Connectionist Grapheme to Phoneme Conversion: Exploring Distributed Representations

Ivelin Stoianov and John Nerbonne
University of Groningen, Faculty of Arts
Email: {stoianov, nerbonne}@let.rug.nl

Abstract

In this paper we explore Simple Recurrent Networks with feature-based letter and phoneme encoding to transform orthographic representations to phonological ones of Dutch words, which is a part of the bigger, text-to-speech synthesis problem. Besides addressing cognitive plausibility, this model performs better than earlier implementations with orthogonal data encoding, which allows useful implementations. We also studied the performance of the network functionally, which led to insights about its behaviour and its implicit linguistics, which in turn were used to present the data to the network during training in a way that would improve learning.

1 Introduction

Converting orthographic word representations to phonological ones is interesting from both cognitive and linguistic points of view. From the former perspective, we are looking for a biologically plausible explanation of a part of our *cognitive* capacity to speak, in particularly the process of reading aloud. On another hand, computational linguistics is still looking for *efficient* methods for text-to-speech synthesis.

Computational linguistics still uses mostly the classical symbolic approaches to this task (Bouma, 2000), which however do not lead cognitive explanations. Connectionism, with its biologically more plausible structures and methods provides cognitively more acceptable alternatives that attract ever increasing attention. Yet, different connectionist models and implementations differ in their plausibility. For example, the first connectionist implementation of such a system – *NETtalk* – by Sejnowski and Rosenberg (1987) uses the static Multilayered Perceptron (*MLP*) (Rumelhart et al., 1986), which is not inherently designed to process dynamic data. However, most of the processes in natural language are dynamic – they span time – which calls for dynamic neural networks.

One neural network model that is widely accepted as useful for linguistic problems is the Simple Recurrent Network (SRN), by Elman (1990). This model is capable of sequential processing because it has a global distributed memory, and the network reaction at each time step depends both on the current input data and the internal memory (see Fig. 1 and Section 2).

Connectionist modelling is not that trivial, because each task can be implemented in a number of ways. The correct choice of the network structure, data encoding and different training parameters determine the outcome of the implementations. For example, the same problem of Grapheme-to-Phoneme Conversion (GPC) is some times modelled with static neural networks and sometimes with dynamic ones. Further, accepting that dynamic networks are the better choice, we again see different data encodings and presentations to the network. For example, Stoianov et al. (1999) used the SRN model with an input consisting of words presented one letter at a time, while Plaut (1999) fed the same network with 10 letters simultaneously. The letters and phonemes in both works were orthogonally encoded.

While those earlier experiments resulted in acceptable performance, there is still a room for improvement. Therefore, we continued our research in the direction of presenting some biasing information to the network, in the form of distributed feature-based input and output representations. This is more plausible than the orthogonal encoding because the distributed data representations are more reliable and cognitively more plausible. As we will see later

in the paper, this also led us to better performance with a smaller network, which in turn is interesting from a practical point of view: it increases the efficiency of the method.

The fine structure of the brain is still difficult to look at and therefore alternative connectionist models of different cognitive processes compete in claims for similarity to the corresponding brain structures. The claims are tested by looking for functional similarities to human performance obtained in psycholinguistic experiments. This also brings insights into the neural networks being used, which are notoriously difficult to explain. Section 4 focuses on this problem. We will go even further there, by drawing some linguistic conclusions related to the structure of the phonemes and syllables on one hand and by studying how different linguistic factors influence the network performance, on the other.

Finally, using those findings, we present in section 5 a new strategy at network training that improves the performance even more.

1.1 State of the Art

In this section, we will briefly present the state of the art in the connectionist modelling of the reading aloud process.

Undoubtedly, one of the most influential works on this subject is the Plaut et al. (1996) paper, where the authors present a study on the quasi-regularity of the grapheme-to-phoneme mapping of approximately 3000 monosyllabic English words. They explored the MLP and the so-called *Attractor model*, which features an extra recurrent layer that searches for the best output pattern matching the initial hidden layer suggestion. However, this model is still a static network, since the transformation orthographic input – phonologic output is performed at once. The input and output layers encode the words with a positional scheme, in which all onsets, nuclei and codas from the training corpus are represented in the orthographic input and phonological output layers. This approach has some obvious limitations such as fixed limits on consonant cluster length and a limitation to single syllables. Nevertheless, this study demonstrates the capacity of a single network to learn a task which incorporates both regular and irregular transformations.

Zorzi et al. (1998) performed similar experiments on the same problem, but they could not handle the irregularity of the mapping with a standard MLP, and therefore they invented an extra set of connections from the input layer to the output layer, which they claim is effectively a *dual-route* neural network. Dual-route models in the symbolic GPC modelling were introduced in (Coltheart et al., 1980), attempting to solve various psycholinguistic phenomena. Such models include a rule-route that transforms regular and unseen words, and a lexicon route that handles all learned words, including words with exceptional pronunciation. However, we argue in (Stoianov et al., 1999) that the Zorzi's architecture is better regarded as a functional view of the network rather than as an effectively new model, since the claimed set of connections could be modelled with a standard MLP.

Stoianov et al. (1999) and Plaut (1999) shifted the focus from static to dynamic networks, by using the SRNs on this problem. In the former, the words were presented to the network one letter at a time (see the next section for details). The latter model used a more specific encoding: the words there were presented to the network in a shifting window containing: the letter to be pronounced; two letters to the left of it; seven letters to the right, and the last phoneme to be pronounced, all of them orthogonally encoded. The output layer contains the orthogonal encoding of all phonemes and the position of the next grapheme to be pronounced. This type of data presentation improved the performance: the Plaut (1999) model, with such a rich input learned the mapping almost perfectly, while the network in (Stoianov et al., 1999) mislearn 10% of the words, although with the acceptable 1.4% phonemic error. The networks in both models exhibited good frequency, consistency and word length effects (see Sect. 4 for details).

1.2 SRNs

Simple Recurrent Networks have the following structure (see Fig.1): Input data (sequences) are presented to the *input* layer, one token at a time. The purpose of the input layer is just to feed the *hidden* layer through a weight matrix, which in turn copies its activations after every step to a *context* layer. The context layer is used to provide another input to the hidden layer – information about the past. And since the acti-

vation of the hidden layer depends on both its previous state (the context) and the current input, the SRNs theoretically are sensitive to the entire history of the input sequence. However, practical computational limitations restrict the time span of the influence of the context information at time t to some 5-10 time steps ahead. In turn, the neurons from the hidden layer output signal through another weight matrix to the neurons from the *output* layer, which in turn is interpreted as a network product.

The network is trained with a supervised training algorithm, which implies two working regimens – a regimen of training and regimen of network use. In the latter, the network is given sequential input data; it reacts according to its knowledge encoded as strengths of weights and its reaction is used for the task at hand. The training regimen comprises a second, training step, during which the network reactions are compared to the desired ones, and the difference is used to adjust the network behaviour in a way that improves the network performance the next time it experiences the same input data.

The particular algorithm used to train the SRNs was the Backpropagation Through Time learning algorithm (Haykin, 1994; Stoianov, 2000). It works both in time and space: the network reaction to a given input sequence is compared to the desired target sequence at every time step and when the whole sequence is processed, the resulting error is propagated back through space (the layers) and time. This results in much faster training than the original simple backpropagation learning algorithm used by Elman (1990) when he introduced the SRNs.

2 Grapheme to Phoneme Conversion with the Simple Recurrent Networks

The method presented in this work uses the SRNs as a neural sequential predictor (Stoianov, 2000 draft). However, in contrast to the standard predicting scheme (e.g., in phonotactics modelling), the output domain (phonology) here differs from the input domain (orthography) and the specifically set sequential mapping guarantees that at every time step only one phoneme will match the current input and context entered so far. In turn, since at any time only one token is permitted to be active, truly distributed representations can be used to encode the output tokens. This facilitates the learning process by providing background knowledge about the nature of the task, thus allowing the same problem to be learned with networks with smaller weight space than earlier, when localistic encoding was employed (Stoianov et al., 1999).

2.1 Distributed Representations

Input and output tokens are encoded with vectors of activations, where each element (neuron) stands for one feature. Different data-encoding schemes determine the concrete functionality of the network.

In the most often used – *orthogonal* – encoding, each neuron n_i stands for one input token c_i, thus, the level of activation of each neuron n_i represents the likelihood $p(c_i)$ that the correspondent token c_i is active. The interpretation (decoding) of this encoding usually follows the *winner-takes-all* rule, which says that the token whose corresponding neuron is most active is the outcome of the system. Since the activations of every neuron are independent each other, this scheme is very useful to represent a set of likelihoods that the correspondent tokens are active in response to the input, which was used in the sequential neural predictor (Stoianov, 2000 draft). However, this encoding is memory expensive, since it needs K neurons to represent K tokens. Also, this encoding is not resistant to noise in data, system damage, etc. Therefore, if the task allows, a *feature-based* representation is better to be used.

Situations allowing feature-based representations are those in which only one token may be a product of the network, for example, in the associative tasks. The networks there have to respond to the input with a specific output pattern (static or sequential). The GPC task, in fact, is exactly a sequential association, thus, permitting distributed representations. As noted above, the GPC was implemented in the framework of the SRNs as a special case of a sequential predictor that requires only one token to be predicted.

As for the feature sets used to represent the tokens, a GPC task prompts for phonemic features. The output phonemes can be encoded according to the specifications of the International Phonetic Alphabet (IPA). It represents

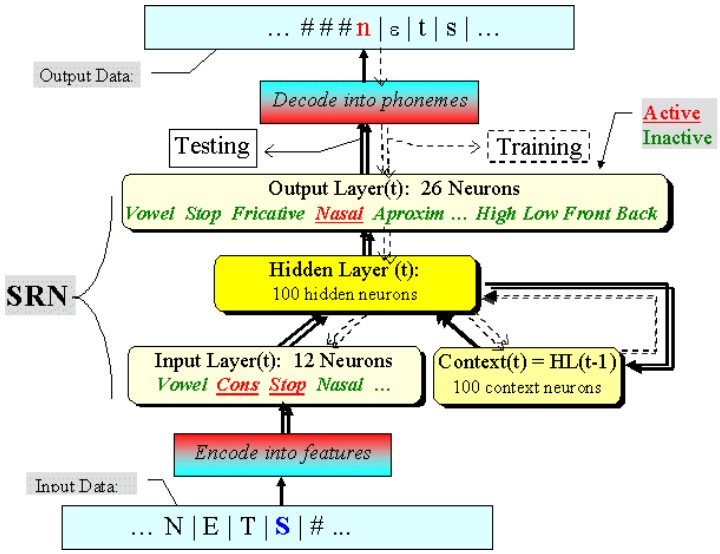

Figure 1: Sequential transformation from orthographic to phonological representations with SRNs and feature-based grapheme and phoneme representations. Words are presented to the input sequentially, one letter at a time. Phonological representations are also produced sequentially, one phoneme at a time, but with 3 steps' delay.

the phonemes with an articulatory set of features, thus making such an encoding biologically plausible since it encodes the phonemes with properties related to their pronunciation. The original feature set in the IPA contains some 40 features, of which about a third are redundant for the most of the European languages and particularly for the Dutch language. Therefore, the feature set we used to encode the phonemes consists only of 25 features, as given in (1).

$$\begin{aligned} &stop, fricative, nasal, approximant,\\ &lateral, trill, voicless, syllabbic, vowel,\\ &bilabial, labio\text{-}dental, dental, alveolar,\\ &palatal\text{-}alveolar, palatal, velar, glottal,\\ &high, upper\text{-}midd, lower\text{-}midd, low,\\ &front, back, round, long \end{aligned} \tag{1}$$

We might choose different strategies for choosing the input feature set, for example, based on visual or linguistic properties. A feature set providing helpful background linguistic information can use an even smaller subset from the IPA feature set and encode the letters with *pro-phonemic* features. Encoding orthographic input with linguistic features is a step-back from cognitive plausibility, but it increases the efficiency of the system, which is important from implementational point of view. The feature set that we used is given in (2). It contains 11 features and provides sufficient feature overlap when the 26 letters are encoded. Feature overlap in distributed representations, in turn, is the source of that bias information.

$$\begin{aligned} &vowel, consonant,\\ &stop, nasal, approximant, voicless,\\ &low, high, back, labial, coronal \end{aligned} \tag{2}$$

In addition, there is one more feature – *delimiter* – used in both the input and output encodings that signals signals for the end of the processed sequences (sequence delimiter).

The symbol encoding process is straightforward – if a given token c_j is to be encoded, its feature-vector is obtained from a look-up table and set to the input/output layer. The decoding process is similar. Since only one phoneme at a time is allowed to be produced by the SRN, the one whose feature vector most closely matches the current output is selected as a product of the network. For those who prefer to stick entirely

to connectionism, one more layer can transform the output representations into localistic encoding.

2.2 Right Context

As discussed, it is important in the Grapheme-to-Phoneme Conversion problem (GPC) to ensure that the learning task requires the network to activate one phoneme only, or at least that for every word and for every time step, the training material does not contain conflicting target spellings (phonemes). We need this in order to make the network spell the words correctly when they contain irregularities, such as (3) in the Dutch language. In this example, the letter sequence "*oe*" is pronounced in two different ways, depending also on the partial right context.

$$foei[fuj] \text{ and } foet[fø{:}t] \tag{3}$$

Providing partial right context is the solution, which could be implemented in different ways. Plaut (1999) provided this by using the simultaneous presentation of 10 letters in a shifting window.

A solution that we chose (Stoianov et al., 1999) was to delay the spelling of the words with d steps, which allowed the network to look d steps right context ahead when producing the phonological representations. This was achieved by training the network on the following sequential mapping (4):

$$\begin{aligned}(C_1^o C_2^o \dots C_{|W_O|}^o \# \dots \#) \Rightarrow \\ (\#_1 \#_2 \dots \#_d C_1^p C_2^p \dots c_{|W_P|}^p)\end{aligned} \tag{4}$$

where '#' represents a *delimiter*; C_i^o stand for the input orthographic tokens and C_j^p for the output phonemic ones.

However, there is a small trap here, in cases where two or more letters are pronounced as one phoneme, such as in (3). This concerns especially polysyllabic words where not only the network might run out of right context, but also it might produce phonemes before the corresponding letters are entered. Since a pronunciation is required after each letter and since some pronunciations can not be predicted until two letters have been seen, the look-ahead buffer might eventually be exhausted. For such cases, an extra mechanism should take care of artificial gaps at the output. In our experiments, we provided 3 letters delay, which turned out to be enough for our training data.

3 Experiment

The proposed model was tested on Dutch monosyllables. Even though monosyllabic words do not represent the entire word space, they do contain most of the complexity of the GPC transformation rules because syllables are the main carrier of the transformation complexity. We used all $5,800$ Dutch monosyllabic words as found in the CELEX lexical database (CELEX, 1993). Among those words there is a number of foreign words, mostly from English and French origin, whose pronunciation differs from that of the regular Dutch words. Yet, in order to simulate a near-real language situation, those foreign words were not filtered out from the database, which makes the task even more difficult. This set was further split into two parts: a training subset L_M^1 containing 4.800 words and a testing one L_M^2 with 1.000 monosyllables – to test the generalisation capacity of the network. The CELEX database contains information about the frequency of the words, which was also used.

The Simple Recurrent Network used had 100 hidden neurons. The input and output layers had 12 and 26 neurons, correspondingly – according to the size of the feature sets used to represent the graphemes and phonemes. The network was trained on the training set and then tested on the testing set, to study its generalisation.

3.1 Training

The training process was organised in epochs, in the course of which all words from the training data set were presented to the SRN according to the logarithm of their frequency ($\overline{f} = 2.2, \sigma = 1.1; min = 1; max = 8$). This decreases the total number of word presentations while preserving the important differences in frequency, thus stressing the most important words and leading to fewer errors on them. The total number of word presentations in one epoch was about $12,500$. For every training sequence, the BPTT learning algorithm was applied. After the training on each epoch, the network performance was evaluated on the same training set, by measuring the number of words and phonemes mispronounced.

Error (%) / Data	L_M^1	L_M^2
Phonemic, Freq.	0.9	1.4
Phonemic, No Freq.	1.37	2.08
Word, Freq.	4.8	6.7
Word, No Freq.	8.5	11.2

Table 1: General SRN performance on the training (L_M^1) and the testing (L_M^2) data sets, measured at phonemic and word level, each of them weighted or unweighted with the word frequency.

The network converged in performance at about the 10^{th} epoch, which is about half as much training time was needed when orthogonal encoding was used (Stoianov et al., 1999). As usual, the network started with a sharp error drop to about 4-5% phonemic error, which slowly decreased to about 1%.

3.2 General Performance

The network was evaluated with two types of error measurement: at a *phonemic* level and at *word* level. The first measures the total number of mispronounced phonemes and the latter one counts the words with at least one mispronunciation. Further, both types of errors were weighted with the correspondent frequencies, which gives an idea how the network would perform in a real-world environment.

We are interested in the network performance on both the training and the testing set. The first one is used to evaluate the network after every training epoch and gives a general idea how the network performs. The performance on the testing set unseen during the training evaluates the generalisation capacity of the model.

Table 1 shows the general network performance. The performance of the network with the feature-based encoding – 0.9% phonemic error and 4.8% word error – is better than the network performance in our previous experiments with orthogonal data encoding, where a SRN with 200 hidden neurons, that is, four times more weights, resulted in 1.2% frequency weighted phonemic error for the training set, and 1.4% phonemic error on the testing set.

We conclude that the distributed encoding is better both for faster training and for the smaller size of the network, but also for its better performance.

4 Evaluation

Although improved, the performance of the network is far from perfect. Preliminary work on polysyllabic words is even worse, with about $15 - 20\%$ erroneous word performance. This raises the question what prevents the network from reaching near-human performance.

Increasing the hidden-layer size in theory increases the network learning capacity, but here it did not lead to improved performance. On the contrary, setting the hidden layer size to some 300-500 neurons worsened the results while increasing significantly the training time. This is because the complexity of the weight space increases significantly and the learning algorithm finds it more difficult to find the solution.

Another approach at improvement is to study the performance of the network by varying properties of the data, analysing where the network makes mistakes and focusing the training on those difficult sequences. Parameters that were expected to influence the performance are the frequency of the words and the regularity of their pronunciation (word consistency). This approach is also interesting from another point of view. In psycholinguistics, different tests study human performance on linguistic tasks, and it is interesting to compare the outcomes of those experiments with the network performance.

4.1 Frequency

Figure 2 shows the performance of the network for three frequency categories – rare, average frequency and frequent words. The network erred more than twice as often on medium-frequency words as compared to high-frequency words, and about twice as often again on rare words. This pattern follows the frequency with which the words were given to the network during training, and we can explain it with the amount of evidence the network was given for the correspondent input-output pattern.

Humans are found to behave similarly in the word naming task, both in terms in performance and reaction time (Fiez et al., 1999). This means that the basic computational principles used in this connectionist model have cognitive justification, at least from a functional point of view. Architecturally, exploiting distributed data representation and processing, by now they

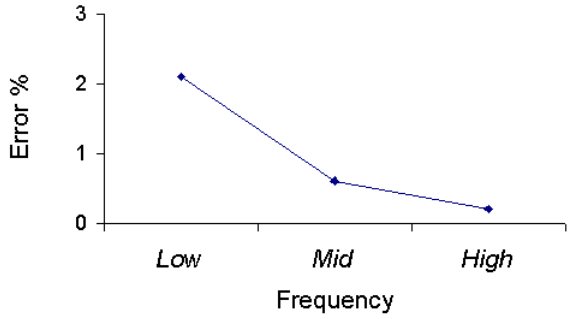

Figure 2: SRN error versus word frequency.

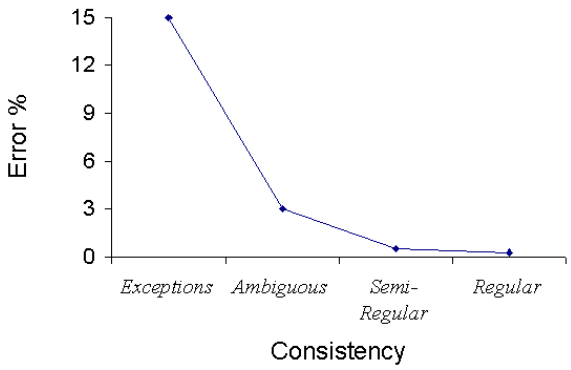

Figure 3: SRN error versus word consistency.

are also the most cognitively plausible models.

4.2 Consistency

The amount of evidence for certain conversions comes not only from the frequency of the particular word that represents it, but also from the number of words that look similar and are pronounced in a similar way, that is, the *consistency* of the orthography-to-phonology mapping for this pattern. Put in another way, consistency measures how much the pronunciation of a given word is like to the pronunciation of orthographically similar words.

Measuring consistency is not trivial. It involves a measure of similarity between words, which is a problem by itself. In our earlier work on this task we measured consistency by matching the sub-syllabic elements *onset*, *nucleus* and *coda* (Stoianov et al., 1999). In that paper we describe in detail how to measure the consistency. As a result, words were assigned a continuous measure with mean value ($23 \pm \sigma = 14$), ranging ($-70 \ldots 80$), which we further split into four categories: *exceptions*, *ambiguous*, *semi-regular* and *regular*.

The variation of the network performance with regard to the word consistency is shown in Fig. 3. As expected, the SRN fails much more often on exceptional words than on regular ones, since the latter "support" each other in the different pattern groups. Humans performed similarly in the word-naming task, making almost no errors on regular words and mispronouncing some exceptional words (Fiez et al., 1999). However, the network performed worse on exceptional words.

Two lessons can be derived from this analysis. Firstly, since the model follows the trends found in the word naming task, we can interpret this as another source of confirmation that the model follows the structural organisation of the human brain.

Secondly – a practical conclusion – since the model under-performed on exceptional words, this means that there is a room either for improvement of the learning strategy, or that the the dual-route idea should be considered as plausible. Other connectionist models managed, indeed to learn similar transformations with single models (Plaut et al., 1996), but those models used as few as half the number of words used in the current experiment. Using fewer words is possible, but one of the targets here is to learn to pronounce all (monosyllabic) Dutch words, not just an easier subset of them. Therefore, the learning should be improved or the architecture should be extended.

4.3 Word Length and Error Position

Dynamic processes are also affected by dynamic properties of the data, e.g., word length and distribution of predictions in time, which are also reflected in psycholinguistic experiments on word naming (Spieler and Balota, 2000 in press). In those examinations a reliable interaction between word length and performance was found: the longer the words, the longer it takes to pronounce them. A similar well-known dynamic property is the performance of human memory on memorising list items. Earlier and later items are remembered best, which results in a U-shaped performance curve.

The network error distribution as a function of word length (shown on Fig. 4) tells us that

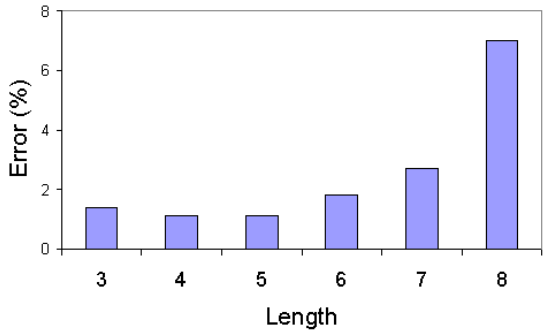

Figure 4: SRN error versus word length

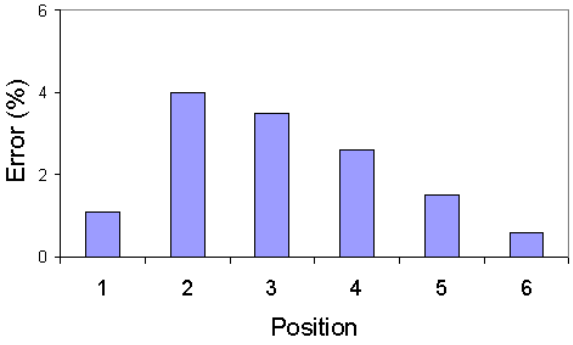

Figure 5: SRN error versus error position.

the network makes many more errors on longer words, which is in parallel to the findings on humans. We can explain this effect in the SRNs with the specific context error that they gain as time progresses.

Error does not linearly correlate with position (Fig.5) – the network makes mistakes in the first half of the words rather than towards the end. A closer analysis reveals that the network makes more mistakes at the time step when the vowel from the nucleus should be spelled out, what we also call a *syllable break*. We found this also in our earlier experiments on phonotactics learning (Stoianov and Nerbonne, 2000) and mapping from orthography to phonology (Stoianov et al., 1999), which we interpreted as a hint for the following structure in the syllable (5):

$$(onset - rhyme(nucleus - coda)) \qquad (5)$$

4.4 Phonemes

It is also interesting to know how well different phonemes are produced; if there are phonemes which are easier or more difficult for the network. In order to study this, the network error was calculated for each phoneme.

Phoneme	Error(%)	Frequency
h	0.0	219
b	0.0	402
v	0.0	207
Delim	0.1	20015
p	0.4	936
r	0.6	1575
x	0.8	755
m	1.0	576
t	1.0	3029
n	1.2	751
s	1.7	2210
f	1.7	475
l	2.0	1303
ŋ	2.1	193
k	2.1	1388
d	3.0	296
ʊ	5.0	484
j	14.4	201
ʃ	43.1	137
ʒ	100.0	20
g	100.0	15

Table 2: SRN error for phonemes-*consonants*. The second data column represents the frequency of the correspondent phonemes in the dataset.

In general, vowels (Table 3) cause more troubles to the network than consonants (Table 2). There is a tendency among the consonants toward larger error for more sonorant consonants (with the exception of 'ʃ'), which leads to the conclusion that the more *sonorant* the phoneme, the larger the error.

The third column in both tables represents the number of occurrences of the corresponding phoneme. For the very infrequent phonemes, both vowels and consonants, the network performs poorly. Hence, the second conclusion is that the lower the phonemic $frequency$, the larger the error. This fact is related again to the amount of evidence the network experiences during learning.

Next, looking at Table 3, one can observe the tendency of the long vowels to produce larger errors than the short vowels (with the exception of

Phoneme	Error(%)	Frequency
ø	0.0	190
y	0.0	190
a:	1.5	401
I	1.8	456
e:	2.8	386
ɔ	3.9	565
ɑ	4.4	755
ʉ	4.4	295
ɛ	5.8	825
y:	6.3	63
i:	7.3	565
o:	7.7	326
u:	8.4	443
ø:	9.5	116
ɔ:	36.4	11
ɛ:	88.9	18
ə	100.0	1

Table 3: SRN Error for phonemes-*vowels*. The second data column gives the phonemic frequency.

'a:' and 'e:', and 'ə', which is rare in monosyllables). This is an interesting finding, from which we might hypothesise that long vowels have more *inconsistent* grapheme-to-phoneme mapping. Indeed, if we search for the orthographic representations of some of those phonemes, we will find that the long vowels stem from a larger variety of orthographic patterns than the short ones. For example, the phoneme 'ø:' is the vowel pronounced in the Dutch words "deuk", "fohn" and "foet". The vowel 'u:' has even more source patterns: "boet", "blues", "tour", "crew" and "croon". On another hand, the short vowel 'I' is pronounced in words such as "blin" and "gym" and the vowel 'ɔ' comes only from words such as "bos".

We can continue in this vein and study the the type of patterns the consonants come from. For example, the consonant 'ʃ' is the pronunciation of as many as six orthographic combinations: "bad*g*e", "bat*ch*", "*ch*eck", "*sh*op", "*sj*aal" and "*tj*ok" and in Table 2 we see that it is associated with a large error. Therefore, the *variety* of letters that match one phoneme is another predictor of the network faulting in this task.

Feature	Examples	Error(‰)
syllabic		0,0
trill	r	0,4
lateral	l	0,9
nasal	m n ŋ	0,9
approximant	w ʋ j	1,9
vowel	...	1,9
Delimiter	#	2,1
stop	...	2,4
fricative	...	3,6
voiceless	...	3,6

Table 4: SRN error / Consonantal *Manner* Features

Feature	Examples	Error(‰)
dental		0,0
glottal	h	0,0
bilabial	p b m	0,6
labio-dental	f v ʋ	1,1
palatal	j c ŋ	1,3
velar	k g x	2,1
alveolar	t d n r l s z	3,5
palatal-alveolar	ʃ ʒ	3,8

Table 5: SRN error / Consonantal *Place* Features

4.5 Phonetic Features

We will complete the study on erroneous SRN performance by examining the error for various phonetic features. In order to facilitate the reading of the data, the features are split into place and manner features, and vowel and consonant features (tables 4,5,6). Further, the features in each group are ordered by the error size.

Feature	Examples	Error(‰)
lower	a: ɑ	2,5
low-mid	ɛ ɔ:	2,8
high	i y u	3,0
front	...	3,0
back	...	3,3
upper-mid	...	3,4
round	...	1,8
long	...	2,7

Table 6: SRN error / Vowel *Place* and *Manner* Features

The most immediate observation is that the size of the phonemic group the corresponding features represent is proportional to the network error. If a given feature is active in a smaller group of phonemes, then the correspondent neuron learns its task more easily than the case of more even phonemic space sampling. For example, the feature *stop* has larger error than the feature *lateral*.

The phenomena of larger error for more balanced features shows that balanced patterns are more difficult to learn than unbalanced ones, which has good theoretical explanation in the framework of informational theory, where a measure for the balance of a certain feature in a given distribution of patterns is called *entropy* (Mitchell, 1997). In this particular case, the entropy $Entr_{f_i}(P)$ of the set of experienced phonemes P with respect to a feature f_i is (6):

$$Entr_{f_i}(P) = -p_{f_i} log_2(p_{f_i}) - p_{\widehat{f_i}} log_2(p_{\widehat{f_i}}) \quad (6)$$

where p_{f_i} is the proportion of the phonemes in the observed phonemic set P which feature f_i, and $p_{\widehat{f_i}} = 1 - p_{f_i}$ is the proportion of the other phonemes in P. Notice, that the entropy is close to zero for more unbalanced distributions and close to one otherwise. The effect of data frequency is also implicitly included here, represented in the set of phonemes P observed by the network during the training.

One interpretation of the entropy in information theory is the number of bits needed to encode an arbitrary pattern – the larger the entropy, the more bits are necessary. On the other hand, we found in the neural networks framework that the larger the entropy of the phonemes with respect to a given feature, the larger the network error, which results in a nice correspondence between the entropy and the difficulty the network meets when trying to learn how to activate this feature. Following this finding, we can predict that the same error pattern will be found in psycholinguistics, which also might help us to explain the way the phonemes are represented in the brain.

We finish with the remark that the previously noted difference in performance on vowels and consonants was found here, too, viz. that in generally the vowels generate larger error and hence, the vowel-related features produce larger error, too.

Error (%) / Data	L_M^1	L_M^2
Phonemic, Freq.	0.49	1.18
Phonemic, No Freq.	0.73	1.78
Word, Freq.	2.62	5.60
Word, No Freq.	3.94	8.71

Table 7: General performance of SRN whose training emphasised inconsistent words. Error on the training (L_M^1) and the testing (L_M^2) data sets is given, measured at phonemic and word level, each of them weighted or unweighted with the word frequency.

5 An improved training method

Now, having the knowledge of how those different factors influence the network performance, it is time to take an advantage of it. For example, the fact that consistency most strongly affects model performance might be compensated for by emphasising more inconsistent words during the network training, that is, presenting them more often to the network in one training session.

To implement this, a second training frequency was computed for each word, inversely proportional to the consistency of that word. Those new frequencies ranged from 1 to 10, with mean value of $2.23, \pm\sigma = 1.15$, which is similar to the original frequency values. Then, the network was trained on the L_M^1 data set, with all other parameters unchanged, until error convergence.

The network performance on the training and testing set is given in Table 7. When compared to the performance of the network trained in the previous conditions (Table 1), the network here errs twice as few as when tested on the training set L_M^1 and shows slight improvement on the testing set L_M^2. Given the fact that the testing words should be considered as realistic *non-words*, it should be expected that the network would perform better on regular words and would not know how to map unseen inconsistent words, converting them by following the GPC "rules" it has learned in training. Since the network has no knowledge of the exceptionally pronounced words that this testing set contains, it generalises, which is registered by the testing procedure as erroneous pronunciation. Therefore, this method would be most advantageous,

if the training is done with a training corpus that is as complete as possible.

The same idea might be extended even further, by emphasising other groups of words that are more difficult to learn. This, we expect, would improve the performance even more.

6 Discussion

Studying the nature of the orthography-to-phonology mapping of the Dutch monosyllabic words and improving the connectionist methodology for its learning were the main objectives of this research. We continued our previous work on this problem by using the more natural distributed representations of letters and phonemes, which led us to a better model.

The same Simple Recurrent Network with twice as small hidden layer (the main processing units) and four times fewer connections (long-term memory) learned the same task even better, with 1.4% phonemic and 8.5% word error. If we weight this performance with the frequency of occurrence of those words in the language, the performance shows 0.9% phonemic and 4.8% word error.

Symbolic methods still perform better. A recent work on a Dutch polysyllabic database achieved 99% phonemic and 92.6% word accuracy, using a combination of hand-crafted rules and transformation-based learning (Bouma, 2000). Our initial results on this data, which we did not discuss in this paper, are much worse, with some 15-20% word error. But symbolic methods do not explain the way humans work with languages, which is the other main goal in connectionist modelling.

In order to find the reasons the network have difficulties in learning this complex mapping, we also studied the type of errors the network makes. This showed some specific error patterns also found in various psycholinguistic experiments (Fiez et al., 1999).

The best known effect is error and naming latency interaction with word frequency – the more frequent the words, the faster they are pronounced and the fewer mistakes are made. Another very important factor that influences the human's performance is the regularity (consistency) of this mapping for each word. What was found in the above and other studies is that the more regular the words are in their pronunciation, the faster and more accurate the responses are. The networks in our experiments performed similarly, which is an evidence for the cognitive plausibility of this architecture, from a functional point of view.

Studying the variation of the error with respect to the phonetic features used to encode the phonemes, we also found the interesting phenomena that the more evenly a given feature partitions the phonetic space, the larger the network error is, which can find a good explanation in information theory with the measure called *entropy*. Based on this finding, we predict that the same error pattern will be found in various psycholinguistic tasks related to phoneme articulation.

Unlike in our earlier work (Stoianov et al., 1999), in this study we did not address the naming latencies, because they behave similarly to the pronunciation accuracy. We studied only accuracy and in Section 4 we showed that they follow the tendencies noticed above.

People still perform better than the network did. One reason for this might be selective attention to those more difficult words. Suggested by the specific pattern of worse network performance for inconsistent words, we applied a similar idea by arranging the training data in a way that would stress the training to the more "difficult" words. This improved the training significantly, by decreasing the SRN error to 0.73% phonemic and 3.94% word error.

Connectionist modelling provides space for continuous improvement. As we just saw, two design details – data encoding and presentation – brought significant improvement to the performance, by more than 50%. And there is still a lot to be done. Further work on this project and more details can be found in (Stoianov, 2000 draft).

References

Gosse Bouma. 2000. A finite state and data oriented method for grapheme to phoneme conversion. In *1st Conf. of the North American Chapter of the Association for Comp. Linguistics, Seattle, WA.*

CELEX. 1993. The celex lexical data base (cd-rom). Linguistic Data Consortium. http://www.kun.nl/celex.

Max Coltheart, K. Petterson, and J.C. Mar-

shall. 1980. *Deep Dyslexia*. Routledge and Kegan Paul, London, Boston.

Jefrey L. Elman. 1990. Finding structure in time. *Cognitive Science*, 14:213–252.

Julie A. Fiez, David A. Balota, Marcus E. Raichle, and Steven E. Petersen. 1999. Effects of frequency, spelling-to-sound regularity, and lexicality on the functional anatomy of reading. *Neuron*, 24:205–218.

Simon Haykin. 1994. *Neural Networks*. Macmillian Publ, NJ.

Thomas Mitchell. 1997. *Machine Learning*. McGraw Hill College.

D.C. Plaut, J. McClelland, M. Seidenberg, and K. Patterson. 1996. Understanding normal and impaired word reading: Computational principles in quasi-regular domains. *Psychological Review*, 103:56–115.

D.C. Plaut. 1999. A connectionist approach to word reading and acquired dyslexia: Extension to sequential processing. *Cognitive Science*, 23:543–568.

D.E. Rumelhart, G.E. Hinton, and R.J. Williams. 1986. Learning internal representations by error propagation. In David E. Rumelhart and James A. McClelland, editors, *Parallel Distributed Processing - Explorations of the Microstructure of Cognition, Volume 1, Foundations*, pages 318 – 363. The MIT Press, Cambridge, MA.

T.K. Sejnowski and C.R. Rosenberg. 1987. Parallel networks that learn to pronounce english text. *Complex Systems*, 1:145–168.

Daniel H. Spieler and David A. Balota. 2000, in press. Factors influencing word naming in younger and older adults. *Psychology and Aging*.

Ivelin P. Stoianov and John Nerbonne. 2000. Exploring phonotactics with simple recurrent networks. In Frank van Eynde, Ineke Schuurman, and Ness Schelkens, editors, *Computational Linguistics in the Netherlands, 1998*, pages 51–68, Amsterdam, NL. Rodopi.

Ivelin P. Stoianov, Laurie Stowe, and John Nerbonne. 1999. Connectionist learning to read aloud and correlation to human data. In *21 Annual Meeting of the Cognitive Science Society, Vancouver, Canada*, pages 706–711, London. Lawrence Eurlbaum Ass.

Ivelin P. Stoianov. 2000. Recurrent autoassociative networks: Developing distributed representations of hierarchically structured sequences by autoassociation. In L. R. Medsker and L. C. Jain, editors, *Recurrent Neural Networks*, pages 205–241. CRC Press, New York, USA.

Ivelin Peev Stoianov. 2000, draft. *Connectionist Lexical Modellinig*. Ph.D. thesis, University of Groningen.

Marco Zorzi, George Houghton, and Brian Butterworth. 1998. Two routes or one in reading aloud? a connectionist dual-process model. *Journal of Experimental Psychology: Human Perception and Performance*, 24/4:1131–1161.