

PHONEME RECOGNITION WITH STAGED NEURAL NETWORKS

Fabio Arciniegas and Mark J. Embrechts (embrem@rpi.edu)

Department of Decision Sciences and Engineering Systems.
Rensselaer Polytechnic Institute, Troy, NY 12065, USA

ABSTRACT

This paper presents a staged series of artificial neural networks (ANNs) for phoneme recognition for text-to-speech applications. Contrary from much of the prior published literature this approach is not restricted to monosyllabic words or the pronunciation of single multi-syllabic words, but can readily be embodied in a program that allows for the reading of a complete text. Also, it does not require pre-processing to align the letters and phonemes on the training data. The training data utilized are the 2000 most common words in American English. As an illustration it is shown that the staged neural neural network approach works excellent for a sample text (in this case the first paragraph of Frank Baum's "The Wonderful Wizard of Oz").

I. Introduction

This paper describes a staged neural network approach that allows for text-to-speech processing for entire texts. This work is novel in the sense that multiple neural networks are used to achieve text to phoneme conversion, can be applied to entire texts, and no pre-processing to align the letters and phonemes is required. This paper also illustrates that the single route approach can be successfully applied for this purpose. The staged networks are trained based on the 2000 most common words in American English. The paper addresses various options for the encoding of the problem and the selection of the proper window size and window focus.

Previous work by Colheart, Curtis, Atkins ([7]), Glusko ([8]), Bullinaria ([4], [5], and [6]) and Sejnowski's NetTalk ([12]) have explored key elements of text-to-speech process which relate also to the study of dyslexia. In their research effort, they recognized and described two processes that have to occur in the human brain in order to read a set of characters (words) and produce a set of phonemes (speech). The first process describes the match between letters and phonemes (phoneme conversion rules) and the second process deals with the conversion process from phonemes to speech.

In order to read it has been discovered that the human brain uses sets of semantic rules or routes. Two lines to study for these semantic rules have been proposed: the single and dual semantic route. The idea behind these rules is that a skilled reader can choose between two different procedures (i.e., lexical and nonlexical routes) for converting text to speech: a dictionary lookup procedure and a letter-to-sound rule procedure.

Most prior studies focused on the single rather than the dual route, because modeling and implementation is easier for the single route. However, it has been found that for special words (e.g., the so-called non-words such as wuff or words with a special pronunciation such as beau) and past tenses (e.g., see [6] and [7]), the single route alone is not sufficient to model the relationship between letters and phonemes. However, Seidenberg and McClelland [11] have challenged this approach, showing that using a single route (lexical) might be enough to read complex words.

In both cases (single and dual route), the main effort has been how to extract and formulate the phoneme sequence for any possible word or text in general. In this paper staged neural network models are developed for the single route phoneme recognition for text-to-speech applications. While other approaches have shown promise (e.g., see Bullinaria [6]) these models are still lacking robust generalization for entire texts.

II. Issues Related to Text-to-Phoneme Conversion

Basically, the approach to the letter-phoneme mapping is based on three fundamental features: recognition, alignment, and context.

Recognition relates to the fact that the same letters and phonemes occurring in different word positions should be recognized as being the same. Not to do so is not only computationally inefficient but also reduces the generalization capabilities as well [6].

With respect to alignment, the key issues relate to distinguishing breaks inside the phrase, the order of the phonemes inside a word, and the matching between letters and phonemes. Two different approaches have been discussed to address how to read an English phrase. A first approach considers the entire phrase, shows it to a neural network, and comes up with the appropriate sequence of phonemes. A second approach (the one followed in this work) breaks down a given phrase in words, shows the network one word at a time and pastes the resulting phonemes in one big file. In this approach “blank spaces” are used to induce a silence period of a distinct duration.

With respect to the matching between letters and phonemes, this paper follows the methodology of phoneme matching by “windowing” the word following Bullinaria [6] or Seidenberg [11]. Bullinaria [6] discusses how the window choice relates to the alignment. A first issue addresses the proper choice for the window size in order to accommodate any long-range dependencies. For example, for the word “THOUGH”, if only a five-space window is chosen, as shown in Fig 1, the last letter will be lost (Phoneme /o/ is giving by “OUGH”).

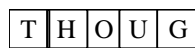


Fig. 1 Five-Window Representation for phoneme /o/

A second issue deals with the fact that if the window size were very large, many units and connections would be vastly underutilized because of the empty window spaces. The use of a series of recurrent connections in lieu of a sliding window has been addressed as well by Bullinaria [5], but did not offer any improvement in performance.

Finally the pronunciation of a particular string of letters depends on the context. Hence, this context information should be used to generate a set of pronunciation rules that will allow a realistic generalization of unknown words and non-words [5].

III. The Staged Neural Network Model

In order to implement a neural network model for text-to-phoneme translation several issues needed to be addressed as outlined in this section. First, a unique representation of inputs (letters) and outputs (phonemes) has to be chosen. Second the training set has to be carefully selected in order to avoid inconsistencies. If the training sample contains “complex” words, it has been shown that overtraining can occur. For example, in the word “ANXIETY”, the “N” corresponds to the phoneme “NG”, which in the rest of the cases correspond to the letters “NG” (i.e., BANG = “/B AH NG/”).

The grapheme-phoneme approach [5] was followed for letter to phoneme matching. The Carnegie Mellon Pronouncing Dictionary (CMPD) was utilized for this purpose [15]. CMPD is a publicly available web-based machine-readable pronunciation dictionary for North American English that contains over 100,000 words and their phonetic transcriptions. The current CMPD phoneme set contains 39 phonemes, for which the vowels may carry lexical stress: 0 for no stress, 1 for primary stress, and 2 for secondary stress. For the basic neural network training set, these features were not included. A 40th phoneme was added in this study to represent the blank or punctuation.

The 2000 most commons American English words [16] were selected for developing a text-to-phoneme staged neural network. A first attempt using the 500 most common English words [14] led to poor network learning because some letter-phoneme matches were not well enough represented in this sample.

Two different representations for the input representation were attempted: a continuous and categorical letter representation. In the first representation, the input for a letter in a particular window position is coded as a continuous number according to:

$$\text{Window}_i = (\text{Code}(\text{Char}) - 64) / 64.$$

Char is the letter that corresponds to the i^{th} place of the window and “Code” is a function that returns the ASCII value for each letter of the alphabet (from 65 to 90, with a value of 64 for blank spaces). One obvious obstacle to this letter

representation scheme is that for the first five letters of the alphabet, A – E, which represent one third of all the phonemes, have values between 0.015 to 0.078, which might be too small to be significant to the net. A discrete categorical representation assigns to each window space a set of 26 binary values (each one for each letter of the English alphabet).

The words are sliced starting with the first letter in the middle of the window, until the whole word has been passed by. This means that the number of pattern per word will be equal to number of letters on the word. An example for “ABLE” is shown in Fig. 2.

Window													Phoneme Output	
1	2	3	4	5	6	7	8	9	10	11	12	13		
						A	B	L	E				EY	
					A	B	L	E					B	AH
				A	B	L	E						L	
			A	B	L	E								

Fig 2. Moving Window for “ABLE”

For the phoneme output representation 40 categorical values were used (one for each phoneme and the 40th neuron represents the blank space). Output values were encoded as 0.1 or 0.9 (to avoid saturation problems related with the sigmoid transfer function). Initially single feedforward neural network models (with one or two hidden layers) were trained on 80 % of the 2000 words resulting in a training set of 10,251 patterns when all the possible window positions are considered. The output layer consists of one or two slabs of 40 neurons representing the phonemes depending on whether the output consists of one or two phonemes corresponding to one single letter in the input window.

For the set of phonemes used in this study it was found that for 0.83% the samples, the match between letters and phonemes was not one-to-one but one-to-two. Some examples of these cases are shown in Table 1.

Table 1. Two-Phoneme Case Examples

Phoneme		Examples
B	AH	A(B)LE, DOU(B)LE, HUM(B)LE
D	AH	BUN(D)LE, HAN(D)LE
G	AH	ANM(G)LE, SIN(G)LE
K	AH	ARTI(C)LE, BYCY(C)LE
P	AH	A(PP)LE, EXAM(P)LE, PEO(P)LE
S	AH	CA(ST)LE, WHI(ST)LE
T	AH	BA(TT)LE, BO(TT)LE, CAS(TT)LE
W	AH	O(N)E, O(N)CE, USU(A)L, USU(A)LLY
Y	AH	CALC(U)LATE, POP(U)LAR
Z	AH	PU(ZZ)LE
G	Z	E(X)ACT, E(X)AMPLE, E(X)IST
K	S	A(X)E, BO(X), E(X)SPENSE, E(X)TRA
Y	UH	C(U)RE, C(U)RIUOS
Y	UW	EXC(U)SE, (U)NION, RESC(U)E

A first approach to implement the two phoneme cases is to consider two 40-phoneme slabs in the output layer (resulting in a total of 80 output neurons): one slab for the first phoneme and a second slab for the second phoneme (if present at all). However, no successful networks resulted from this approach (i.e., overall error was around 77%, but none of the two-phoneme cases was really recognized). Hence, it was decided to follow a staged neural network approach where we first identify dual phoneme cases in the first neural network stage. In the second stage two different neural networks are used to deal with one and two-phoneme cases separately.

IV. Results

The stage which recognizes whether we one is dealing with a one or two-phoneme case uses two categorical output neurons with outputs [0 1] or [1 0] depending on the outcome. For these models we tried the categorical and continuous

encoding for the input letters. The results for the discrete encoding were by far better than those from the continuous input encoding (as can be seen in Table 2). Different window sizes were tested as well. Centered window encoding is able to capture all the single phoneme cases and 57% of the two phoneme cases. The best network employed a five-position size window and two hidden layers (43 and 67 neurons).

Looking at the different patterns for the two phoneme cases (See Table 1), it was found that to predict the two phoneme-cases it was not necessary to employ the entire window length. All that is required is just the space before the letter that is being considered in addition to the three following spaces. For example, to predict the phonemes that corresponds to the second “U” (“U” = /Y UW/) in the word “USUALLY” or the “B” in “ABLE” (“B” = /B AH/), the information shown in Fig. 4 is enough.

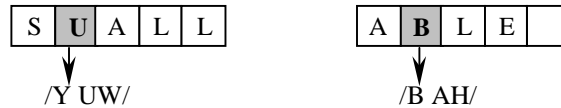


Fig. 4. Window Structure for the Two Phoneme Case Matches. The space in bold corresponds to the letter that wants to be predicted.

The kind of window shown in Fig. 4 which we refer to as second position asymmetric windowing was very successful in the recognition of one or two phoneme-type cases. It recognized all of the one-phoneme cases and 95 percent of two-phoneme cases (only 6 cases out of 182 were not recognized by the network). The results for the different neural networks for the first stage addressing one or two-phoneme recognition are shown in Table 2 (The (*) indicates that the net configuration used is as the one showed in Fig 4.).

Table 2. First stage Summary Results for Recognizing One or Two Phoneme Matches

Net Configuration							Error	
ID	Case Input	Window Size	Input Layer	Hidden Layers		Output Layer	Total	Two Phon
				1	2			
1	D	5	130	43		2	99%	52%
2	D	5	130	43	67	2	99%	57%
3	C	7	7	11		2	87%	36%
4	D	7	182	43	67	2	99%	28%
5	C	13	13	11		40	99%	0%
6	D	13	338	43	67	40	99%	0%
7	C	5 (*)	5	11		2	99%	67%
8	D	5(*)	130	43		2	99%	67%
9	D	5(*)	130	43	67	40	99%	94%

For the second stage (the phoneme-matching stage) neural networks with a structure similar to these of the first stage were utilized (except that now we have 40 rather than 2 output neurons). Only categorical letter encodings were considered for this stage. Generally window sizes of 5 spaces were used but a seven-window structure was tested as well (but no improvements resulted). For larger windows than 7 the results were generally poorer. For the two-phoneme case the best performing neural network model consisted of one hidden layer with 43 neurons (94% recognition). Actually, the only pattern of the validation set that the net was not able to recognize was the word “HUMAN”. The “U” corresponds to the phonemes /Y UW/ but the net output was /Y UH/. The results are shown in Table 3.

For the one-phoneme cases we experimented with centered windows of 5, 7, 9, 11 and 13 spaces (all of the categorical encoding type). We also tried out second position asymmetrical windows of 5, 6 and 7 spaces such as the one shown in Fig 4. Consistent with the two-phoneme case larger windows did not lead to better results. The best neural network utilized a second position asymmetric window with 5 spaces (using the categorical input encoding) and two hidden layers with 43 and 67 neurons respectively. The total error for this network is 87%.

It was found that for the 13% (260 of 2002) of the cases that were not recognized, only six of them were consonants (i.e., /D/ and /T/ were confused with /DH/ and /TH/). Also, it was found that for all the missed cases the net was activating two phonemes at the same time: the target and the closest vowel phoneme. For example, if the output target

is /UH/, the net activated /UH/ and /UW/. This error was most notorious in the case of the phonemes related with /I/ (/IH/, /IY/ and /EY/).

Table 3. Summary Results Two-Phoneme Case

Net Configuration							Total Error
ID	Case Input	Window Size	Input Layer	Hidden Layers		Output Layer	
				1	2		
1	D	5	130	43		40	67%
2	D	5	130	43	67	40	57%
3	D	5(*)	130	43		40	94%
4	D	5(*)	130	43	67	40	94%
5	D	7(*)	156	43		40	53%
6	D	7	182	43		40	48%
7	D	7	182	43	67	40	38%
8	D	13	338	67		40	14%
9	D	13	338	43	67	40	24%
10	C	13	13	11		40	62%

Also, for some of the examples of interest cited by Bullinaria [6] such as “TALKED”, “WALKED” and “THOUGH”, there was no misclassification.

Finally to have an idea of how well the staged neural network approach would perform on a blind data set we actually fed the neural network the first paragraph of Frank Baum’s “The Wonderful Wizard of OZ’ as shown in Figure 5. The phonetic results are also shown and indicates someobvious mispronunciations. However the results clearly show that the staged neural network approach for phoneme recognition is promising.

W	THE	CYCLONE																														
O	DH	AH	S	IH	K	L	OW	N																								
NN		AH		IH	K	L	OW	N																								
W	DOROTHY	LIVED	IN	THE	MIDST	OF	THE	GREAT	KANSAS																							
O	D	AO	R	AH	TH	IY	L	AY	V	D	IH	N	DH	AH	M	IH	D	S	TH	AH	V	DH	AH	G	R	EY	T	K	AE	N	Z	AH
NN		ER		AH	DH	IY		AY	V		EH	N		AH		IH	D	S		AH	V		AH		R	EH	T		AE	N	S	AH
W	PRAIRIES	WITH	UNCLE	HENRY	WHO	WAS	A	FARMER	AND																							
O	P	R	EY	R	IY	Z	W	IH	DH	AH	NG	K	L	HH	EH	N	R	IY	HH	UW	W	AA	Z	AA	F	AA	R	M	ER	AE	N	
NN		R	EH	R	IY	Z	W	IH	TH	AH	NG	K	L	HH	EH	N	IY		HH	UW	AH	EY	Z		J	AA	R	M	ER		N	
W	AUNT	EM	WHO	WAS	THE	FARMERS	WIFE																									
O	AE	N	T	EH	M	HH	UW	W	AA	Z	DH	AH	F	AA	R	M	ER	Z	W	AY	F											
NN		N	T		M	HH	UW	AH	EY	Z		AH	J	AA	R	M	ER	S	W	AY	F											
W	THEIR	HOUSE	WAS	SMALL	FOR	THE	LUMBER	TO	BUILD																							
O	DH	EH	R	HH	AW	S	W	AA	Z	S	M	AO	L	F	AO	R	DH	AH	L	AH	M	B	ER	T	UW	B	IH	L	D			
NN	DH	IH	R	HH	AW	Z	W	EY	Z	S	M	AH	L	F	AO	R		AH	L	AH	M	B	ER	T	UW	B	IH	IH	L	D		
W	IT	HAD	TO	BE	CARRIED	BY	WAGON	MANY	MILES	THERE																						
O	IH	T	HH	AE	D	T	UW	B	IY	K	AE	R	IY	D	B	AY	W	AE	G	AH	N	M	EH	N	IY	M	AY	L	Z	DH	EH	R
NN	IH	T	HH	AE	D	T	UW	B	AH	AA	R	IY	D	B	AY	AH	AH	G	AH	N	M	AE	N	IY	R	AY	L	S		EH	R	
W	WERE	FOUR	WALLS	ONE	ROOM	AND	THIS	ROOM	CONTAINED																							
O	W	ER	F	AO	R	W	AO	L	Z	W	AH	N	R	UW	M	AE	N	D	DH	IH	S	R	UW	M	K	AH	N	T	EY			
NN	W	EH		F	AO	R	W	AO	L	S	W	AH	N	R	UW	M	AE	N	D	DH	IH	S	R	UW	M	K	AH	N	T	EY		
W	A	RUSTY	LOOKING	COOKSTOVE	A	CUPBOARD	FOR	THE																								
O	AA	R	AH	S	T	IY	L	UH	K	IH	NG	K	UH	K	S	T	OW	V	AA	K	AH	B	ER	D	F	AO	R	DH	AH			
NN	AA	R	AH	S	T	IY	L	UH	K	IH	NG	K	UH	K	S	T	UW	V	AA	K	AH	B	AO	ER	D	F	AO	R		AH		
W	DISHES	A	TABLE	THREE	OR	FOUR	CHAIRS	AND	THE	BED																						
O	D	IH	SH	AH	Z	AA	T	EY	B	AH	L	TH	R	IY	AO	R	F	AO	R	CH	EH	R	Z	AE	N	D	DH	AH	B			
NN	D	IH	SH	EH	Z	AA	T	EY	B	AH	L	TH	R	IY	AO	R	F	AW	R	CH	EH	R	S	EH	N	D		AH				

Figure 5. Frank Baum’s The wonderful wizard of Oz and it’s phonetic encoding as predicted by the staged neural net approach.

V. Conclusions

This paper outlined a successful text-to-speech conversion methodology with a two-stage neural network. This staged approach first recognizes whether one is dealing with a one or two phoneme case and then does the phoneme matching in the second stage. Even though there are still a few obvious mispronunciations with this approach it is expected that the performance will improve with further experimentation.

VI. References

- [1]. Adamson, M. J., Damper, R. I. "A Recurrent Network that Learns to Pronounce English Text". *Proceedings of 1996 International Conference of Spoken Language Processing, ICSLP'96*. Vol. 4, pp. 1704 - 1707.
- [2]. Beale, R., Finlay, J. "Neural Networks and Pattern Recognition in Human-Computer Interaction". Ellis Horwood Workshops, 1992.
- [3]. Bullinaria, J. A. "Internal Representations of a Connectionist Model of Reading aloud". *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*. Pp. 84-89, 1994.
- [4]. Bullinaria, J. A. "Neural Network Learning from Ambiguous Training Data". *Connection Science*, Vol. 7, 99-122, 1995.
- [5]. Bullinaria, J. A. "Modeling Reading, Spelling, and Past Tense Learning with Artificial Neural Networks". *Brain and Language* 59, 236-266, 1997.
- [6]. Colheart, M., Curtis, B. & Atkins, P. "Models of Reading Aloud: Dual-Route and Parallel-Distributed-Processing Approaches". *Psychological Review*, 1992.
- [7]. Glusko, R.J. "The Organization and Activation of Orthographic Knowledge in Reading Aloud". *Journal of Experimental Psychology: Human Perception and Performance*, Vol. 5, 674-691, 1979.
- [8]. Norris, D. "A Quantitative Multiple-Levels Model of Reading Aloud". *Journal of Experimental Psychology: Human Perception and Performance*, Vol. 20, pp. 1212-1232, 1994.
- [9]. Patel, M. Using Neural Nets to Investigate Lexical Analysis". *PRICAI'96: Topics in Artificial Intelligence*. Pp. 241-252, 1996.
- [10]. Seidenberg, M., McClelland, J. "A Distributed, Developmental Model of Word Recognition and Naming". *Psychological Review, Volume 4* 1989.
- [11]. Sejnowski, T.J. & Rosenberg, C.R. "Parallel Network that Learn to Pronounce English Text". *Complex Systems*, Vol. 1, No. 1, p. 145-168.
- [12]. Van Santen, J., AT&T Bell Laboratories. "Assignment of Segmental Duration in Text-to-Speech Synthesis". *Computer Speech & Language*, Vol. 8, No. 2, April 1994.
- [13]. members.tripod.com/educ8u/500words.html: 500 hundred more common words in American English
- [14]. www.speech.cs.cmu.edu/cgi-bin/: The CMU Pronouncing Dictionary. Carnegie Mellon University.
- [15]. http://www.uri.edu/comm_service/cued_speech/1000most.html. CUE practice with the 1000 most common words.
- [16]. <http://www1.harenet.ne.jp/~waring/Wordlists/vocfreq.html>. The Word Frequency Lists.