A Norwegian NETtalk Program

Terje Kristensen
The Departement of Computer Science ,
Bergen University College,
Nygårdsgaten 112,
N-5020 Bergen, Norway.
E-mail: tkr@hib.no

Bernd Treeck
EMISOFT
Bergen HighTech Centre
Thormøhlens gate 55,
N-5008,Bergen , Norway.
E-mail: bernd.treeck@emisoft.no

Ronny Falck-Olsen (ronnyfo@student.hin.no)

Abstract

This paper describes a Norwegian NETtalk program, a neural network program which learns to convert Norwegian text to phonemes. The original database consists of about 60000 Norwegian words and their transcriptions. This phoneme database is developed by the Norwegian Telecom and is further developed to be used in this program. The Sampa notation for Norwegian is used for the phoneme transcriptions. The experiments have been performed on a PC-platform and on only a sample of words from this database.

Keywords
Norwegian NETtalk, Sampa, Backpropagation, Phoneme recognition

1. Introduction

We are all experts in reading and communicating in our native language. We often tend to forget how long it took to acquire these skills when we have become good at them. The problem of pronouncing written Norwegian text illustrates many of the features of skill acquisitions and expert performance. When we read aloud, the letters and words are first recognised by our visual system from images on the retina. The information encoded visually is then transformed into information how to produce correct speech sounds. In this paper we are concerned with the transformation of Norwegian letters to the earliest representation of phonemes before the pronunciation of them. The problem of converting text to speech has traditionally been approached with rule-based knowledge representation, such as natural language translation. The neural network approach or the connectionist approach is another one that has been popular in recent years [1,2,6]. Such an approach emphasises the importance of the connections between the nodes in a neural network to solve problems of natural language. In two earlier papers [4,5] we have demonstrated that a neural network is fully capable of automatically hyphenating Norwegian.

In this paper we describe a Backpropagation neural network that learns to transcribe different Norwegian words. The result of the learning process is a string of phonemes for each word. This string can then be converted to sounds with a digital speech synthesis by a look-up table that stores the different transcriptions. In this paper we only demonstrate a Norwegian NETtalk program that is designed to perform the converting of strings of letters to strings of phonemes. As far as we know, a Norwegian NETtalk program has not been developed before. The NETtalk program of English has been described by Sejnowski and Rosenberg in their classical paper from 1987 [8]. This paper reports a number of experiments in which a neural network was trained on a list of a 1000 very common English words.

2. Backpropagation

In a Backpropagation network the processing elements are organised in layers. Neurons in one layer receive signals from neurons in the layer directly below and send signals to neurons in the layer directly above. Connections between neurons in the same layer are not allowed. Except for the input layer nodes, the net input to each node is the weighted sum of outputs of the nodes in the previous layer. Each node is activated in accordance with the input to the node and the activation of the node. The net input to a node j in a layer is

(1) $S_j = \Sigma\, w_{ji}\, a_i$

where a_i is output from node i in the previous layer. The output of node j is then given by

(2) $a_j = f(S_j)$

where f is usually the *sigmodal* activation function given by

(3) $$f(S_j) = \frac{1}{1+e^{-S_j}}$$

In the learning phase we present patterns to the network and the weights are adjusted so that the produced outputs from the output nodes are equal to the target. In fact, we want the network to find a single set of weights that will satisfy all the (input, output) pairs presented to it.
In general, the outputs $\{a_{pi}\}$ (p is the current pattern) of the nodes in the output layer will not be the same as the target or desired values $\{t_{pi}\}$. The system error or the cost function for the network is defined by:

(4) $$E = \frac{1}{N}\frac{1}{2}\sum_p \sum_i \left(t_{pi} - a_{pi}\right)^2$$

where N is the total number of patterns. A gradient search should be based on minimisation of the expression in equation (4). The weight updating rule then becomes:

(5) $\Delta w_{ji} = - \alpha\, \delta_j\, a_i + \beta \Delta w_{ji}$

where α is the learning rate, δ_j is the error for any node and β is the momentum constant. For a more thorough discussion see reference [3, 9,10].

3. Sampa for Norwegian

The transcription of different Norwegian words will in this paper be based on the Sampa notation [7]. In Sampa for Norwegian the consonants and vowels are classified into different subgroups like:

Consonants
There are six *plosives*. These are given by:

Symbol	**Word**	**Transcription**
p	hopp	hOp
b	labb	lAb
t	lat	lA:t
d	ladd	lAd
k	takk	tAk
g	tagg	tAg

There are six *fricatives*:

f	fin	fi:n
v	vin	vi:n
s	lass	lAs
S	skyt	Sy:t
C	kino	Ci:nu (not syllable-inal)
j	gi	ji:
h	ha	hA:

There are five *sonorant* consonants (nasals, liquids, trills)

m	lam	lAm
n	vann	vAn
N	sang	sAN
l	fall	fAl
r	prøv	pr2:v

Vowels
There are 9 *long* vowels:

i:	vin	vi:n
e:	sen	se:n
{:	vær	v{:r
A:	hat	hA:t
y:	lyn	ly:n
2:	søt	s2:t
O:	våt	vO:t
u:	bok	bu:k
}:	lun	l}:n

and nine *short* vowels:

i	vind	vin
e	send	sen
{	vært	v{rt
A	hatt	hAt
y	lynne	lyne
2	søtt	s2t
O	vått	vOt
u	bukk	buk

}	lund	l}n

There are seven *diphthongs*:

{i	vei	v{i
2y	høy	h2y
A}	sau	sA}
Ai	kai	kAi
Oy	konvoy	kunvOy
}i	hui	h}i
ui	hoi	hui

In addition there are important *allophonic* variants for which the transcription has been agreed:

rt	hardt	hArt	(retroflex t)
rd	verdi	v{rdi:	(retroflex d)
rl	ærlig	{:rli	(retroflex l)
rn	garn	gA:rn	(retroflex n)
rL	blå	brLO:	(retroflex flap)

In cases where the *dental* consonants do not change into *retroflexes*, they are transcribed using the separator sign (ASCII 45), e.g.

r-d	verdig	v{r-di

Norwegian has two contrasting **tonemes**, but only in stressed syllables. Tone 1 is indicated by the ordinary stress mark, Tone 2 by a doubled stress mark, e.g.

stress and toneme 1	bønder	"b2n@r	(peasants)
stress and toneme 2	bønner	""b2n@r	(beans)

In our phoneme database ordinary and double stress marks are indicated by ! and !! instead of " and "" that is used above.

4. Theory

The Backpropagation network that is used consists of 3 or more layers of processing units. The different Norwegian words are fed into the input layer. All the units in each layer are connected to the units in the layer above. The output layer determines the output phoneme. The Norwegian language following the Sampa scheme for transcription consists of 39 phonemes. Only one unit in the output layer is used to represent each phoneme. The output layer will therefore consist of 39 output units.

The neural network is given a certain view to the different words as shown in figure 4.1. In our case the window consists of *seven letters*. At any given time the letter in the middle of the window is active. In figure 4.1 this is r. When r is active, all the other letters will be inactive.

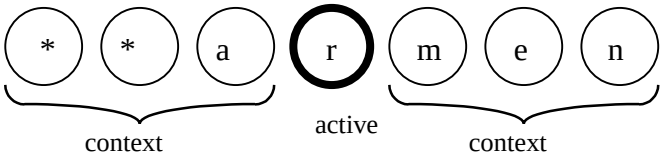

Figure 4.1 The Norwegian word "armen"in a seven-letter window. The associated phoneme of the letter r will then fire in the output layer.

To handle the beginning and end of each word a space is used, indicated by *. * is used to fill up the space of seven letters. The desired output of the network is the correct phoneme associated with the centre or the fourth letter. The other six letters (three on either side of the centre letter) provide the *context*. The words are stepped through the window letter by letter. At each step the network computes the phoneme associated with the letter of the centre. The weights of the network are then adjusted to how close the computed transcription matches the correct one.

The size of the window may vary. By choosing a seven-letter window we may have the significant information needed to correctly transcribe a Norwegian word. A larger window will require more computer power and longer training times of the network. But any size of the window can be chosen in our transcription scheme.

5. Training

The database of Norwegian words with their transcriptions has been developed by the *Norwegian Telecom* at Kjeller. The Backpropagation network was trained on only a sample of words from this training database. The original database contains about 60000 different words. Each pattern in the training database consists of eight lines, seven for the representation of the letter and one for the target. To indicate that the phoneme in the middle of the window fires, a 1 is set for this phoneme in the output layer. A binary code scheme has been developed to represent each letter and its associated phoneme. A program in *Delphi* has been developed to make a binary pattern file of all the input words and their transcriptions.

We first used a learning regime where the learning rate was constant, but it did not work so well. Later on we therefore used a learning regime where the learning rate was changed during the training session, from 0.3 in the beginning to 0.05 at the end. The reason for this was to let the network detect more detailed phoneme structures during learning. For each pass through the training file, the configuration of the network was written to file. In this way we could test the network at any stage during training.
The words in the training database have been trained without and with indication of *stress*. The reason for this was to compare the recognition of the different phonemes when stress is included in the transcription regime or not. A sample of words from the training database are given in table 5.1.

Word	**Transcription**
ape,	A:p@
apen,	A:p@n
apene,	A:p@n@
aper,	A:p@r

```
apet,      A:p@t
ene,       e:n@
ener,      e:n@r
enere,     e:n@r@
enn,       en
enorm,     enOrm
ens,       e:ns
ensa,      ensA
```

Table 5.1 Some words from the training database and their transcriptions.

Table 5.2 shows the binary equivalent of the pattern '****ape**' in the input window. We see that '*' (space) is represented as 16 zeros and that only 'a' is active (1). The letter 'a' is represented by 1 in position 10 of the coding table. The associated phoneme 'A' is indicated by 1 in the target line of the pattern.

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0
```

*Table 5.2 The binary equivalent of the pattern '****ape**'..

The transcription scheme consists of 16 different characters. Each letter is represented by a seven- letter window. The input layer therefore consists of 16 x 7 = 112 neurons. In Sampa for Norwegian there are 39 different phonemes. The output layer or target then consists of 39 neurons as shown in table 5.2. A double t (tt), for instance, will in this regime be replaced by a single one (t).

Two extra neurons have been added in the output layer when stress in the syllables is included. The signs ! and !! are used to indicate *ordinary* and *double stress*. The output layer will in this case consist of 41 neurons. Table 5.3 shows some words in the training file when stress is included.

```
ape,       !!A:p@
apen,      !!A:p@n
apene,     !!A:p@n@
aper,      !!A:p@r
apet,      !!A:p@t
apt,       !A:pt
apte,      !!A:pt@
```

Table 5.3. Some words of the training database with indication of stress .

Table 5.4 shows a binary pattern with indication of stress. The pattern shows *ordinary* stress represented as 1 0 at the beginning of the target line.

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
```

Table 5.4 A binary pattern with indication of stress.

The two extra nodes in the output layer (target) code for ordinary or double *stress*. When a phoneme is fired in the input layer, the neural network program checks if there is *ordinary*, *double* or *no* stress in front of the actual phoneme, by inspecting the first two nodes in the output layer. After the actual word has stepped through the input window, the word and its associated phonemes are written to file.

6. Experiments and results

6.1 Experiment 1

6.1.1 One hidden layer

The experiments have been done on a 500 Mhz Pentium PC with a 128 mbytes memory. A program in C was developed to compare the test file and the results produced by the network. The experiments were first done with a network topology of *one* hidden layer. Different numbers of hidden units were chosen to find the right network topology. The numbers were 20, 30, 40, 50, 60, 70. The global error was estimated in each case. The network learned quickly to associate approximately correct a phoneme with its letter. This took only about 20 iterations to learn. The experiments indicated that the optimum number of hidden units were 50. A network with a lower number of hidden nodes was not able to catch all the different patterns of the phoneme structures so well. In a Backpropagation network with the number of hidden nodes greater than 50 some over-learning took place.

During the test phase the words with their phonetic transcriptions were written to file. In this way it was quite easy to compare the produced result of the network with target.

Without stress The training database consisted in this case of about 1000 Norwegian *five-letter* words. Each word has been constructed from the vowels *a, e, i, o* and the consonants *m, n, p, r, s, t*. A training session of this situation is shown in figure 6.1. In the figure we can see the different layers of the network together with the learning curve. The training time for this particular experiment was about 4 CPU hours.
After the learning session had been finished, the network was tested on words from the training database, that are *known* words for the network . A recognition rate of 83 % was measured. This means that of about 1000 words, the NETtalk program was able to transcribe

830 input words. The target and the produced phoneme string were identical in these cases. However, if we compare the target and the produced phoneme string on a character basis, the recognition rate would be higher.

With stress A similar training database of the same five-letter words was constructed with *stress* included. The recognition rate for this situation was about 75 %. This is a more complex problem for the network to cope with, to identify at the same time the actual phoneme and its stress. So a lower recognition rate was not unexpected in this case.

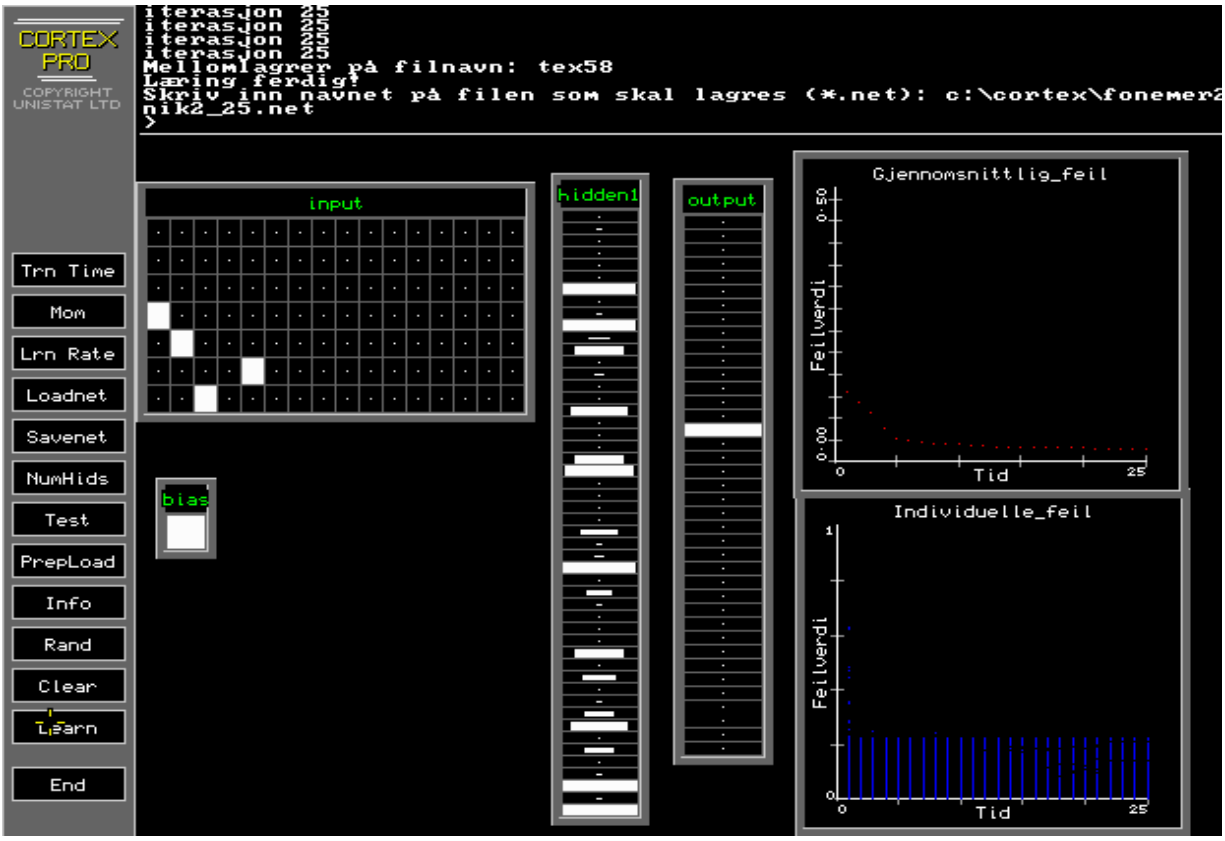

Figure 6.1. A screen dump of the training session. One graph shows the individual errors of the patterns, and one shows the average or global error.

6.1.2 Two hidden layers

Two similar experiments were performed with the same training databases as above with *two hidden layers*. In the first case the first and second hidden layer consisted of 40 and 30 neurons and the second case of 50 and 30. But the results did not improve. We therefore decided to use only *one hidden layer* in the rest of the experiments. The experiments have shown that a neural network with one hidden layer is capable of carrying out transcription of *five-letter* words of the Norwegian language.

6.2 Experiment 2

The experiment in the last paragraph was now extended to a situation where the words in the training database could be of *any length*, but still constructed from the same letters as earlier. The training databases consisted of two separate files of about 3500 words each, without and with stress indication. The training time for this situation was about 11 CPU hours. The recognition rates were 72 % and 63 % respectively.

The results achieved were not very good, but the transcription of Norwegian words of any length is a more complex problem for the network to handle than that of only five letters.

A sample of Norwegian test words and produced words when stress is included look like:

String	Target	Produced
ansees	!Ans@s	!Ans@@s
anser	!Ans@r	!Anse:r
ansiennitet	Ansi@ni!te:t	Ansi@ni!te:t
ansiennitetene	Asi@ni!te:t@n@	Ansi@ni!te:t@n@
assisterte	Asi!ste:rt@	Asi!ste:rt@
assortert	Asurt!e:rt	As}rte:rt
attesterte	At@!ste:rt@	At@!ste:rt@
einer	!!{in@r	!!{in@r
einerne	!!{in@rn@	!!{in@rnn@
emna	!!emnA	!!emnA
entreen	AN!tre:@n	en!tre:@n
representanten	repr@s@n!tAnt@n	repr}s@n!tAnt@n
representantene	repr@s@n!tAnt@n@	repr}s@n!tAnt@n@

7. Conclusion so far

A Norwegian NETtalk program has been developed as a Backpropagation neural network. The neural network has been trained on only a sample of words from a phoneme database developed by the Norwegian Telecom. The transcription scheme is based on *Sampa* notation for Norwegian.

Two main experiments have been performed. In one experiment we have shown that a Backpropagation neural network is fully able to transcribe Norwegian *five-letter* words constructed from the letters *a, e, i, o, m, n, p ,r, s, t* of Norwegian to their equivalent phonemes.

In the second experiment words of *any length,* constructed from the same letters as before, have been studied. The recognition rate was not so good in this case. However, this problem has not yet been thoroughly explored. We believe that an optimal network configuration has not yet been found. The experiment has only been carried out with 50 units in the hidden layer. This number may be too low.

To answer how well a NETtalk program in general is capable of transcribing words of the Norwegian language remains to be fully explored. Such a program must for instance be able to transcribe words constructed from *all* the letters of the Norwegian of any length. Such an experiment is planned to be carried out in the near future.

References

[1] Cawley,G.C., Noakes,P.D. “L.S.P” Speech Synthesis using Backpropagation Networks. In Proceedings of I.E.E International Conference on Artificial Neural Network, ANN-93, Brighton, UK, 1993.

[2] Cawley,G.C., Green,A.D.P.The Application of Neural Networks to cognitive phonetic Modelling. In Proceedings of I.E.E. International Conference on Artificial Neural Network AN-91, Bournemouth,UK, 1991.

[3] Kristensen,T. Nevrale Nettverk, Fuzzy logikk og Genetiske algoritmer. Cappelen Akademisk Forlag, 1997.

[4] Kristensen,T., Langmyrh,D., Treeck,B, Falck-Olsen,R. Two Approaches to Hyphenating Norwegian. In Proceedings of “Norsk Informatikkonferanse”, NIK’98, Tapir publisher, Norway.

[5] Kristensen,T., Treeck,B, Falck-Olsen,R. Hyphenation by an Atificial Neural Network. In Proceedings of “Norsk Informatikkonferanse”, NIK’97, Tapir publisher, Norway.

[6] Rubin,M.G. Artificial Neural Networks for Speech Analysis/Synthesis. Chapman and Hall, 1994.

[7] The Sampa notation (http://www.phon.ucl.ac.uk/home/sampa/norweg.htm)

[8] Sejnowski,T.J., Rosenberg,C.R. Parallel Networks that Learn to Pronounce English Text. Complex Systems Publications Inc, 1987.

[9] Taylor,J.G. Neural networks. Alfred Waller Limited, Publishers. London, 1995.

[10] Yoh-Han.P. Adaptive Pattern Recognition and Neural Networks. Addison-Wesley, 1989.