Automation
Computers
Applied Mathematics
ISSN 1221–437X
Vol. 17 (2008) no. 1
pp. 5–11

Continuous Speech Phoneme Recognition Using Dynamic Artificial Neural Networks

Domokos József and Toderean Gavril

Domokos József: • Sapientia University, Electrical engineering department, 540485, Corunca,
Şoseaua Sighişoarei no. 1/C, O.P. 9, C.P. 4
••Technical University of Cluj-Napoca, Communications department, 400027, Cluj-Napoca, Gh. Barițiu street, no. 26-28

jdomokos@com.utcluj.ro

Toderean Gavril: Technical University of Cluj-Napoca, Communications department, 400027, Cluj-Napoca, Gh. Barițiu street, no. 26-28

toderean@pro3soft.ro

Abstract: Phoneme classification and recognition is the first step to large vocabulary continuous speech recognition. This step represents the acoustic modeling part of such a system. In hybrid speech recognition systems phoneme recognition is made by artificial neural networks (ANN's).
The main objective of this paper is the investigation of dynamic ANN's, namely the Time-Delay Neural Networks (TDNN) and Recurrent Neural Networks (RNN) - that are the most suitable for recognition of time sequences. There are presented two types of TDDN's: Focused Time-Delay Neural Networks (FTDNN) and Distributed Time-Delay Neural Networks (DTDNN) respectively and a Layer Recurrent Neural Network (LRNN).
The development of a phoneme recognizer application using dynamic ANN's for OASIS Numbers databases is also described. There are also presented the phoneme classification experiments and the results for the ANN's. Finally some conclusions are drawn based on the experimental results.

Key Words: continuous speech recognition, phoneme classification, dynamic neural networks, OASIS Numbers

Received: September 25, 2008

1 Introduction

A state of the art statistical speech recognition system is based on Hidden Markov Models (HMMs) [2]. Mathematically such a system can be described as fallows: given a set of acoustic vectors (A), after the feature extraction stage, we are searching for the most probable word sequence (W).

$$W^* = argmax_w \{P(W|A)\} \tag{1.1}$$

The above equation can be transformed using Bayes rule as fallows:

$$W^* = argmax_w \{P(A|W) \cdot P(W)\} \tag{1.2}$$

In the above equation the conditioned probability P(A—W) represents the acoustic model and the probability P(W) represents the language model part of the system.

We try to examinate in this paper an alternative way for acoustic modeling instead of Hidden Markov Models, one which is based on artificial neural networks (ANN's). This approach is widely used in hybrid HMM - ANN speech recognition systems, also called connectionist speech recognition [2].

2 Features used for recognition

Our feature extractor module, presented in [4], extracts 39 coefficients, namely 13 Mel Frequency Cepstral Coefficients (MFCC) with their first and second order time differences (delta and double delta parameters), using a signal preprocessing preemphasis of high frequencies with a first order FIR (Finite Impulse Response) filter and a 256 samples length Hamming windowing with 15 % overlap.

3 The OASIS Numbers database

This continuous speech database was developed at University of Szeged, by Artificial Intelligence Research Group for training and testing speaker independent number recognition systems [7].

The database contains 26 short numbers and 20 long numbers each of them uttered two times by a number of 66 speakers

All the short utterances are manually segmented and annotated. These utterances can be used to train the system. The rest of utterances can be used for testing.

The train and test sets we used, were the recommended ones. The phoneme set consists of 32 phonemes marked using SAMPA standard.

4 Artificial neural networks

Neural networks can be classified into static and dynamic categories. Static networks have no feedback elements and contain no delays; the output is calculated directly from the input through feedforward connections. In the case of dynamic networks, the output depends not only on the current input to the network, but also on the current or previous inputs, outputs, or states of the network.

Dynamic networks can also be divided into two categories: those that have only feedforward connections, and those that have feedback, or recurrent connections.

Dynamic networks are generally more powerful than static networks. Because dynamic networks have memory, they can be trained to learn sequential or time-varying patterns. This is why they can be used in speech recognition applications, especially for phoneme recognition.

We have made earlier some experiments for phoneme recognition using static, feedforward neural networks, and in this paper we want to examinate dynamic neural networks for the same task.

5 Focused time-delay neural network (FTDNN)

The most straightforward dynamic network, which consists of a feedforward network with a tapped delay line at the input is called the focused time-delay neural network (FTDNN). This is part of a general class of dynamic networks, called focused networks, in which the dynamics appear only at the input layer of a static multilayer feedforward network [3]. The following figure illustrates the FTDNN which we have used for our experiments.

The network has 39 inputs, one for each MFCC, delta and double delta coefficient. The 32 outputs of the network corresponds to each phoneme from the dictionary. The transfer function of the hidden layer is sigmoidal and for the output layer we have softmax transfer function to consider the output values as probabilities.

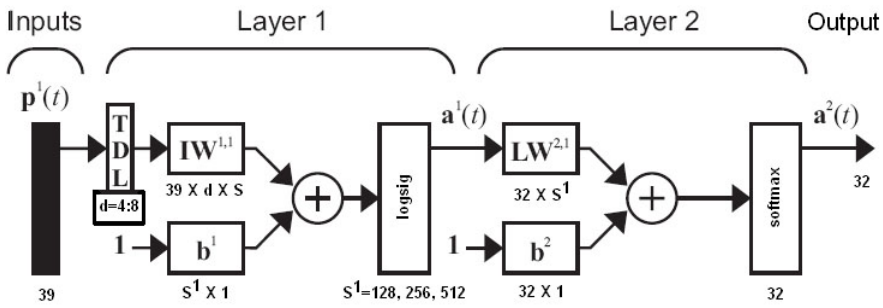

Figure 1: FTDNN architecture, modified after [3]

We apply to the network input the current feature vector and from 4 up to 8 context feature vectors using a tapped delay line (TDL) with d = 0, 1, 2, 3, 4 delay steps. Such the 39x 128 x 32 FTDNN with 4 delay lines (d = 0 : 4) has 39x5x128=24.960 input weights (IW) and 128x32 = 4.096 layer weights (LW). The total number of network weights was 29.056.

6 Distributed time-delay neural network (DTDNN)

The DTDNN has tapped delay line memory not only at the input to the first layer but also distributed throughout the network. The figure 2 shows the DTDNN used in our experiments. The 39x 128 x 32 DTDNN with 4 delay lines (d1 = 0 : 4) for the input layer and 3 delay lines (d2 = 0 : 2) for the hidden layer has 39x5x128=24.960 input weights (IW) and 128x3x32 = 12.288 layer weights (LW). The total number of network weights was 37.248.

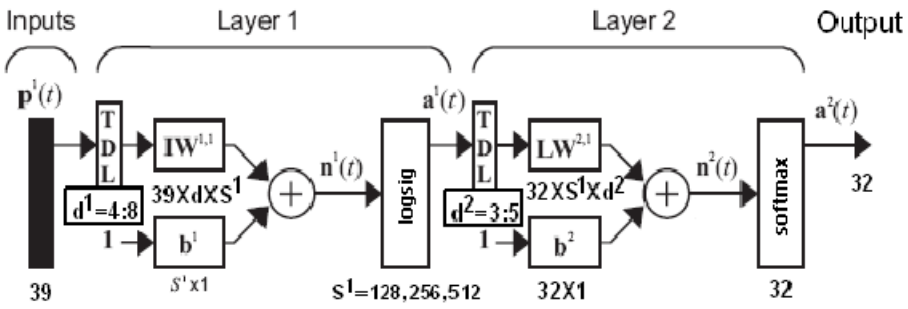

Figure 2: DTDNN architecture, modified after [3]

7 Layer-recurrent neural network (LRNN)

The next dynamic network to be introduced is the Layer-Recurrent Network (LRNN). An earlier simplified version of this network was introduced by Elman [3]. In the LRNN, there is a feedback loop, with a single delay, around each layer of the network except for the last layer. The original Elman network had only two layers, and used a tansigmoidal transfer function for the hidden layer and a purelinear transfer function for the output layer. The original Elman network was trained using an approximation to the backpropagation algorithm. The LRNN generalizes the Elman network to have an arbitrary number of layers and to have arbitrary transfer functions in each layer. In our case we use sigmoidal transfer function for the hidden layer and softmax for the output layer. The training of LRNN is made using the backpropagation algorithm. The following figure illustrates the two-layer LRN we have used for the experiments.

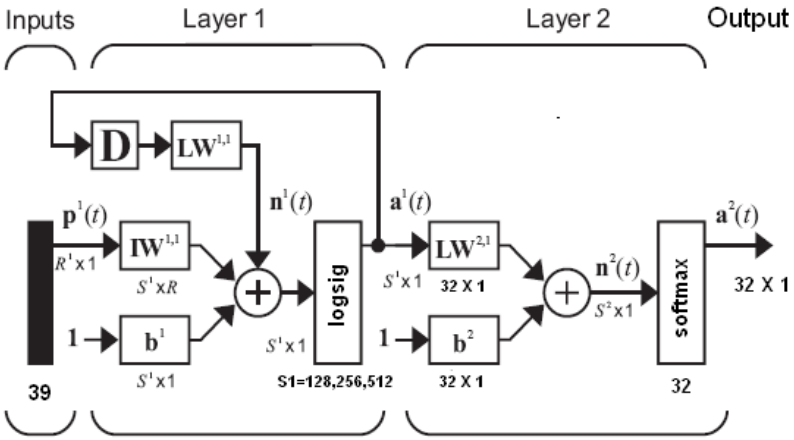

Figure 3: LRNN architecture, modified after [3]

8 Recognition results

Our previous results obtained using a MLP with 39 inputs, 128 computational units in the hidden layer and 32 outputs (39x128x32) are presented in Table 1.

We improve the recognition rates by using contextual information at the network input, but we must increase network dimensions. The recognition result using 4 left context frames and 4 right context frames and the current frame with a 351x128x32 MLP architecture are presented in Table 2.

Train epochs	*Test percentage (%)*
1	57.63
11	64.81
20	67.18
30	67.65
40	68.47
50	69.35

Table 1: 39x128x32 MLP results[4]

Train epochs	*Test percentage[%]*
11	67.74
50	74.06
100	78.22
150	79.69
200	81.63
250	82.93
300	83.48
350	83.77
450	84.25

Table 2: 351x128x32 MLP results[4]

Phoneme	*Total nr. of frames*	*Recognition [%]*	*Misrecognized frames*	*Phoneme*	*Total nr. of frames*	*Recognition [%]*	*Misrecognized frames*
E	2625	79,58	536	e:	913	65,5	315
+	312	64,42	111	u:	293	90,44	28
d'	292	67,12	96	s	738	77,1	169
z	765	75,82	185	u	236	80,08	47
r	322	53,42	150	k	295	85,42	43
h	918	82,79	158	-:	237	18,57	193
A:	953	46,27	512	2:	228	70,61	67
o	561	74,69	142	1	433	24,94	325
m	259	55,6	115	o:	199	62,81	74
O	1053	77,21	240	l:	147	43,54	83
i	592	68,58	186	J	125	43,2	71
n	1324	50,68	653	j	128	64,84	45
-	1252	72,44	345	2	345	75,36	85
ts	680	74,71	172	i:	329	91,49	28
t	525	66,67	175	~	19192	98,47	293
v	492	62,6	184				

Table 3: 351x128x32 MLP architecture results for each phoneme from the dictionary [4]

The recognition results for each phoneme from the dictionary are presented in Table 3. The FTDN we used achieves better result that MLP without contextual information, and the TDNN outperform but not significantly also the FTDNN. See Table 4 and Table 5.

We obtained the best results without contextual information using the LRNN architecture. The recognition results are presented in Table 6.

Train epochs	*Test percentage (%)*
10	58.12
20	66.47
30	68.11
40	69.23
50	71.16

Table 4: FTDNN architecture results

Train epochs	*Test percentage (%)*
10	59.12
20	67.77
30	69.94
40	71.42
50	72.87

Table 5: DTDNN architecture results

Train epochs	*Test percentage (%)*
10	64.86
20	69.47
30	71.94
40	74.39
50	76.12

Table 6: LRNN architecture results

9 Conclusions

The results confirm our expectation that dynamic networks outperform static networks in phoneme recognition task. Using TDNN instead of static network we can slightly improve recognition result, but because of the delay lines the network training takes much time. Recurrent networks performs better than MLP and TDNN.

Using contextual information we can improve the recognition rate, but also increase the input layer of the networks (and such increase the number of weights of network)

In comparison to other GMM and HMM based phoneme classification methods our LRNN model provides fairly good results [1].

The tests made on the OASIS Number database shows us that the application performs well on small and medium size databases. We want to try our system on some bigger databases like TIMIT to compare the achieved results with state of the art phoneme recognizer systems [8].

Our final scope is to use our phoneme recognizer as the acoustic modeling part of a continuous speech recognition system considering the HTK Toolkit. We have developed and presented already the language modeling part of the system in [5].

References

[1] M. Antal. *Phoneme recognition for ASR* Proceedings of the 6th International Conference COMMUNICATIONS, Bucharest, 2006

[2] H. Bourlard, N. Morgan. *Connectionist speech recognition* Kluwert Academic Publishers, 1994.

[3] H. Demuth, M. Beale, M. Hagan. *Neural network toolbox ^{TM} 6 user's guide.* Mathworks Inc., 2008.

[4] J. Domokos, G. Toderean. *Phoneme classification using MLP.* Inter-ing conference with international participation, Tîrgu-Mures, 2007.

[5] J. Domokos, G. Toderean. *Language modelling on Susane corpus.* Proceedings of the 7th International Conference COMMUNICATIONS, Bucharest, 2008.

[6] A. Graves, J. Schmithuber. *Framewise phoneme classification with bidirectional LSTM networks.* Proceedings of IEEE International Joint Conference on Neural Networks, 2005.

[7] MTA-SZTE, Mesterséges Intelligencia Tanszéki Kutatócsoport. *OASIS Numbers adatbázis* 2002.

[8] T. Robinson *Recurrent nets for phone probability estimation* Proceedings of ARPA Continuous Speech Recognition Workshop, 1992.

[9] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K. J. Lang *Phoneme recognition using time-delay neural networks,* IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 37, Issue 3, pp. 328-339.