A TIME-DERIVATIVE NEURAL NET ARCHITECTURE — AN ALTERNATIVE TO THE TIME-DELAY NEURAL NET ARCHITECTURE

K.K. Paliwal[1]
Speech Research Department
AT&T Bell Laboratories
Murray Hill, NJ 07974

ABSTRACT — Though the time-delay neural net architecture has been recently used in a number of speech recognition applications, it has the problem that it can not use longer temporal contexts because this increases the number of connection weights in the network. This is a serious bottleneck because the use of larger temporal contexts can improve the recognition performance. In this paper, a time-derivative neural net architecture is proposed. This architecture has the advantage that it can utilize information about longer temporal contexts without increasing the number of connection weights in the network. This architecture is studied here for speaker-independent isolated-word recognition and its performance is compared with that of the time-delay neural net architecture. It is shown that the time-derivative neural net architecture, in spite of using less number of connection weights, outperforms the time-delay neural net architecture for speech recognition.

1. INTRODUCTION

Hidden Markov modeling is a popular, and perhaps the most successful, technique today for speech recognition. Its main advantage lies in its ability to model the time variability of the speech signals. However, it has a drawback that it does not provide enough discrimination between classes (such as phonemes and words) as their models are usually obtained using the maximum likelihood algorithm. On the other hand, neural networks, and particularly multilayer perceptrons, provide good discrimination between classes and, hence, are being investigated by many researchers for speech recognition. The major problem with the multilayer perceptrons is that they are restricted to static patterns and it is difficult to extend them to the classification of time-varying speech signals. However, a number of neural net architectures have been recently proposed in the literature (see [1] for references) to overcome this problem. Notable among these is the time-delay neural net architecture proposed by Waibel et al. [2]. Time variability is incorporated in this architecture by utilizing temporal context in the form of time delays. In this

[1]On leave from Computer Systems and Communications Group, Tata Institute of Fundamental Research, Homi Bhabha Road, Bombay-400005, India.

0-7803-0118-8/91/0009-0367$01.00

architecture, the input to a neuron at a given time is computed as a weighted sum of not only the present outputs of the lower layer's neurons, but the outputs from the past as well (i.e., their time-delayed versions). Though the time-delay neural net architecture has been used in a number of applications, such as phoneme recognition [2], phoneme spotting [3, 4] and isolated word recognition [5], it has the problem that the number of connection weights become very large when longer temporal contexts have to be incorporated in the network. Therefore, this architecture can not be used for larger temporal contexts. This is a serious bottleneck because the use of larger temporal contexts can improve the recognition performance [6].

In the present paper, we propose an alternative to the time-delay neural net architecture. We call it the time-derivative neural net architecture. This architecture incorporates the time variability present in the speech signals by utilizing the temporal context in the form of time derivatives, instead of time delays. Here, the input to a neuron at a given time is obtained by taking the weighted sum of the outputs of the lower layer's neurons and their derivatives at that time. This architecture has the advantage that it can utilize information about longer temporal contexts without increasing the number of connection weights in the network. This is not possible with the time-delay neural net architecture, as mentioned earlier.

Some speech recognition experiments are performed in this paper to study the performance of the time-derivative neural net architecture and compare it with that of the time-delay neural net architecture. The recognizer is used here in a speaker-independent mode for recognizing an utterance from a 9-word vocabulary consisting of English e-set alphabets. It is shown in this paper that the time-derivative neural net architecture, in spite of using less number of connection weights, outperforms the time-delay neural net architecture for speech recognition.

The organization of this paper is as follows. The time-derivative neural net architecture is described in Section 2. In Section 3, this architecture is studied for speaker-independent isolated-word speech recognition and performance of this architecture is compared with that of the time-delay neural net architecture. Conclusions are reported in Section 4.

2. THE TIME-DERIVATIVE NEURAL NET ARCHITECTURE

In this section, the time-derivative neural net architecture is briefly described. This architecture uses a feed-forward fully-connected multi-layer perceptron type of neural network. It has one input unit, one output unit and a number of hidden units. However, in the architecture described below, only one hidden unit is used.

We use here the letter j for the index of a neuron in the output layer, i for the index of a neuron in the hidden layer, and k for the index of a neuron in the input layer. We denote the input to a neuron by the letter x and its output by the letter y. Let n_j, n_i and n_k be the number of neurons in the output, hidden and input layers, respectively. The number of neurons in the input layer is equal to the number of features in the feature vector. The number of neurons in the output unit is equal to the number of classes (or, words in the vocabulary). The number of neurons in the hidden layer is selected depending on the complexity of the recognition problem.

Let the speech utterance to be recognized be represented by a sequence of feature vectors,

$$\mathbf{C} = \{\mathbf{c}(1), \mathbf{c}(2), \ldots, \mathbf{c}(T)\}, \tag{1}$$

where $\mathbf{c}(t)$ is the feature vector at time frame t and T is the number of frames in the speech utterance. In order to recognize this speech utterance, the outputs of the neurons in the output layer have to be computed. This is done as follows.

For each time frame t, the components of the feature vector are applied to the inputs of the neurons in the input layer and outputs of these neurons are computed by assuming identity transfer function for each of these neurons. That is,

$$y_k(t) = c_k(t), \tag{2}$$

for $k = 1, 2 \ldots, n_k$ and $t = 1, 2, \ldots, T$.

The input to the i-th neuron in the hidden layer at time t is computed as the weighted sum of the outputs of the input layer's neurons and their derivatives at time t; i.e.,

$$x_i(t) = \sum_{k=1}^{n_k} \sum_{d=0}^{D_k} w_{ikd} y_k^{(d)}(t) + \theta_i, \tag{3}$$

where D_k is the number of derivatives of the outputs from the input layer used for the computation of the inputs for the hidden layer, θ_i is the threshold for the i-th neuron in the hidden layer, and w_{ikd} is the connection weight associated with the i-th hidden neuron, the k-th input neuron and the d-th derivative. $y_k^{(d)}(t)$ is the d-th derivative at time t of the output of the k-th neuron in the input unit and is computed as the d-th orthogonal polynomial coefficient over a finite length window as follows [7, 8, 9, 10]:

$$y_k^{(d)}(t) = K^{(d)}(L) \sum_{l=-L}^{L} a_l^{(d)}(L) y_k(t+l), \tag{4}$$

where $(2L + 1)$ is the window length and $K^{(d)}(L)$ is a scaling constant which depends mainly on window length and order of derivative[2]. $a_l^{(d)}(L)$ is the d-th orthogonal polynomial defined over a window length of $(2L + 1)$ frames. The first few orthogonal polynomials are [7]:

$$a_l^{(0)}(L) = 1, \tag{5}$$

$$a_l^{(1)}(L) = l, \tag{6}$$

[2]For the computation of the 0-th order derivatives, the window length is set to 1 frame; i.e., L=0, and the scaling constant K used in Eq. (4) is set to 1.

$$a_l^{(2)}(L) = l^2 - (L^2 - 1)/12, \tag{7}$$

$$a_l^{(3)}(L) = l^3 - (3L^2 - 7)l/20. \tag{8}$$

The output of the i-th neuron in the hidden layer at time t is computed from its input using the sigmoid nonlinear function as follows:

$$y_i(t) = 1/(1 + \exp(-x_i(t))). \tag{9}$$

The outputs of all the neurons in the hidden unit for all the time frames are computed by using Eqs. (3-9) for $i = 1, 2, \ldots, n_i$ and $t = 1, 2, \ldots, T$.

In a similar fashion, the outputs of the neurons in the output unit are computed as follows:

$$y_j(t) = 1/(1 + \exp(-x_j(t))), \tag{10}$$

for $j = 1, 2, \ldots, n_j$ and $t = 1, 2, \ldots, T$. Here, $x_j(t)$ is the input to the j-th neuron in the output layer and is computed as the weighted sum of the outputs of the hidden layer's neurons and their derivatives as follows:

$$x_j(t) = \sum_{i=1}^{n_i} \sum_{d=0}^{D_i} w_{jid} y_i^{(d)}(t) + \theta_j, \tag{11}$$

where the derivative $y_i^d(t)$ is computed as follows:

$$y_i^{(d)}(t) = K^{(d)}(L) \sum_{l=-L}^{L} a_l^{(d)}(L) y_i(t + l). \tag{12}$$

The outputs of the individual neurons in the output layer obtained from Eq. (10) for $t = 1, 2, \ldots, T$ define the a posteriori probabilities of the words associated with these neurons at different time frames [11, 12]. These can be normalized by their a priori probabilities to get the emission probabilities for time $t = 1, 2, \ldots, T$ [13]. The likelihood of the speech utterance (defined by the sequence C of the feature vectors) coming from the word associated with the i-th neuron in the output layer is obtained by multiplying these emission probabilities as follows:

$$L_i(\mathbf{C}) = \prod_{t=1}^{T} y_i(t)/p_i, \tag{13}$$

where p_i is the a priori probability of the word associated with the i-th neuron in the output layer. The speech utterance is recognized by maximizing the likelihood; i.e., by using the maximum likelihood decision rule [14].

3. SPEECH RECOGNITION EXPERIMENTS AND RESULTS

In this section, the time-derivative neural net architecture is studied for speech recognition and its performance is compared with that of the time-delay neural net architecture. Speech recognition experiments reported in this section are conducted in a speaker-independent mode. Isolated-word speech recognizer is used in these experiments. The vocabulary of the recognizer consists of 9 English e-set alphabets (B, C, D, E, G, P, T, V and Z).

Speech data base used in the recognition experiments consists of 24 utterances per word from 4 speakers (2 male and 2 female) for training, and 40 utterances per word from the same 4 speakers for testing. These utterances are digitized at a sampling rate of 6.67 kHz. An 8-th order linear prediction analysis is performed frame-wise every 15 ms using a Hamming window of 45 ms, and each frame is represented by a feature vector of 12 liftered cepstral coefficients [15].

The time-derivative neural net architecture used in the recognition experiments has 3 layers: the input layer, the hidden layer and the output layer. The numbers of neurons in the three layers are 12, 16 and 9, respectively; i.e., $n_k = 12$, $n_i = 16$ and $n_j = 9$. In these experiments, we set $D_i = 0$. This means that the inputs to the output units are obtained as a weighted sum of the outputs of the hidden neurons and their derivatives are not used. Outputs and their derivatives are used only in the computation of the inputs to the hidden neurons; i.e., $D_k > 0$. For training the neural net, the error-back propagation algorithm is used with the total-squared error criterion [16].

As mentioned earlier, the time-derivative neural net architecture has the advantage that it can utilize longer temporal context without increasing the number of connection weights. This is done here by increasing the window length. In order to see how the longer temporal context improves the recognition performance, we use the time-derivative neural net with $D_k = 1$. This means that the total number of connection weights used in the neural net (including the thresholds) is 553. Performance of this net as a function of the duration of temporal context is shown in Table 1. It can be seen from this table that the time-derivative neural net architecture

Table 1: Recognition performance of the time-derivative neural net architecture as a function of the duration of the temporal context.

Duration of the temporal context (in ms)	Number of connection weights	Recognition accuracy (in %)
75	553	67.2
135	553	72.2
195	553	76.1
255	553	75.0

results in better recognition performance when the longer temporal context is used. Recognition performance is best for the temporal context of 195 ms duration. Note

that Hanson and Applebaum [10] have made similar observations for the hidden Markov model based speech recognizer.

Next, we study the recognition performance of the time-derivative neural net architecture using higher order derivatives. For this, we fix the duration of the temporal context to 195 ms, and study the recognition performance as a function of the number of derivatives. When we say that the number of derivatives is 2 (for example), it means that we are computing the input to a neuron as a weighted sum of the outputs from neurons in the lower layer and their first and second derivatives. Results are shown in Table 2. It can be seen from this table that use of second

Table 2: Recognition performance of the time-derivative neural net architecture as a function of number of derivatives.

Number of derivatives	Duration of the temporal context (in ms)	Number of connection weights	Recognition accuracy (in %)
1	195	553	76.1
2	195	745	78.6
3	195	937	77.2

derivative improves the performance, but at the cost of more number of connection weights. However, when the third derivative is used the recognition performance goes down, in spite of the fact that we are using more number of connection weights. This happens because the amount of data available for training the speech recognizer is limited, a common problem in pattern recognition [17].

Thus, we have seen that the time-derivative neural net based speech recognizer performs better with the longer temporal context. Note that use of longer temporal context does not increase the number of connection weights in the neural net. Also, the inclusion of higher derivatives in the neural net improves the recognition performance, if sufficient amount of training data is available for training.

Now, we compare the recognition performance of the time-derivative neural net architecture with that of the time-delay neural net architecture. For this, we study the performance of the time-delay neural net based speech recognizer for different temporal delays. Here also, the inputs to the output neurons are computed from the outputs of the hidden neurons without using any temporal delays. The temporal delays are used only at the outputs of the input neurons to compute the inputs to the hidden neurons. Recognition results as a function of number of temporal delays are shown in Table 3. It can be seen from this table that we can get better recognition performance by using more number delays in the neural net. However, the performance does not improve after 3 delays, in spite of the fact that we are using more number of connection weights. This happens, as mentioned earlier, due to the limited amount of training data [17]. Note that if we want to use a longer temporal context, we have to use more delays. This means that we have to increase the number of connection weights. This increases the computational cost and memory requirements. Also, use of more connection weights does not always

Table 3: Recognition performance of the time-delay neural net architecture as a function of number of delays.

Number of delays	Duration of the temporal context (in ms)	Number of connection weights	Recognition accuracy (in %)
1	30	553	62.8
2	45	745	63.3
3	60	937	67.3
4	75	1129	67.5
5	90	1321	67.5

improve the recognition performance due to the limited amount of training data.

By comparing Table 3 with Table 2, we can see that for the same number of connection weights, the time-derivative neural net architecture performs much better than the time-delay neural net architecture. Also, comparison of Table 3 with Table 1 shows that the time-derivative neural net architecture can utilize information about longer temporal contexts without increasing the number of connection weights in the network, while this is not possible with the time-delay neural net architecture. The reason for this is that the time-derivative neural net architecture models the temporal context in terms of time derivatives and, thus, does not increase the number of connection weights for the longer temporal contexts. The time-delay neural architecture uses explicit time delays to incorporate the temporal context. Therefore, it has to increase the number of delays (and connection weights) to utilize longer temporal contexts.

It may be noted that we have not used in our recognition experiments the derivatives of the outputs of the hidden neurons to compute the inputs to the output neurons. This will be done in our future research work. Also, note that the present system can be thought of as a single-state hidden Markov model based speech recognizer where outputs of the output neurons are used as emission probabilities. Recognition performance can be improved by using these emission probabilities with the multi-state hidden Markov models, as done by Haffner et al. [18].

4. CONCLUSIONS

In this paper, a time-derivative neural net architecture is proposed for speech recognition. This architecture has the advantage that it can utilize information about longer temporal contexts without increasing the number of connection weights in the network. This is not possible with the time-delay neural net architecture. The time-derivative neural net architecture is studied here for speaker-independent isolated-word recognition and its performance is compared with that of the time-delay neural net architecture. It is shown that the time-derivative neural net architecture, in spite of using less number of connection weights, outperforms the time-delay neural net architecture for speech recognition.

References

[1] H. Bourlard and C.J. Wellekens, "Speech dynamics and recurrent neural networks", in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing* (Glasgow, Scotland), May 1989, pp. 33-36.

[2] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano and K.J. Lang, "Phoneme recognition using time-delay neural networks", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-37, pp. 328-339, Mar. 1989.

[3] M. Miyatake, H. Sawai, Y. Minami and K. Shikano, "Integrated training for spotting Japanese phonemes using large phonemic time-delay neural networks", in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing* (Albuquerque, NM), Apr. 1990, pp. 449-452.

[4] K.J. Lang, A.H. Waibel and G.E. Hinton, "A time-delay neural network architecture for isolated word recognition", *Neural Networks*, vol. 3, pp. 23-43, 1990.

[5] L. Bottou, F.F. Soulie, P. Blanchet and J.S. Lienard, "Experiments with time-delay networks and dynamic time warping for speaker independent isolated digits recognition", in *Proc. Eurospeech*, Sept. 1989.

[6] A.J. Robinson and F. Fallside, "A dynamic connectionist model for phoneme recognition", in *Proc. Eurospeech*, Sept. 1989.

[7] N.R. Draper and H. Smith, *Applied Regression Analysis*. New York: Wiley, 1981.

[8] S. Furui, "Cepstral analysis techniques for automatic speaker verification", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 254-272, Apr. 1981.

[9] F.K. Soong and A.E. Rosenberg, "On the use instantaneous and transitional spectral information in speaker recognition", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp.871-879, June 1988.

[10] B.A. Hanson and T.H. Applebaum, "Robust speaker-independent word recognition using static, dynamic and acceleration features: Experiments with lombard and noisy speech", in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing* (Albuquerque, NM), Apr. 1990, pp. 857-860.

[11] H. Bourlard and C.J. Wellekens,"Links between Markov models and multilayer perceptrons", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, pp. 1167-1178, Dec. 1990.

[12] E.A. Wan, "Neural network classification: A Bayesian interpretation", *IEEE Trans. Neural Networks*, vol. 1, pp. 303-305, Dec. 1990.

[13] N. Morgan and H. Bourlard, "Continuous speech recognition using multilayer perceptrons with hidden Markov models", in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing* (Albuquerque, NM), Apr. 1990, pp. 413-416.

[14] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[15] B.H. Juang, L.R. Rabiner and J.G. Wilpon, "On the use of bandpass liftering in speech recognition", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP35, pp. 947-954, 1987.

[16] D.E Rumelhart, G.E. Hinton and R.J. Williams, "Learning representations by back-propagating errors", *Nature*, vol. 323, pp. 533-536, Oct. 1986.

[17] L.N. Kanal and B. Chandrasekaran, "On dimensionality and sample size in statistical pattern classification", *Pattern Recognition*, vol. 3, pp. 225-234, Oct. 1971.

[18] P. Haffner, M. Franzini and A. Waibel, "Integrating time alignment and neural networks for high performance continuous speech recognition", in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing* (Toronto, Canada), May 1991, pp. 105-108.