Non-linear Prediction of Speech Signal Using Artificial Neural Nets

K. Ashouri, M. Amini and M.H. Savoji
Electrical and Computer Engineering Faculty
Shahid Beheshti University
Evin Square, Tehran 1983963113, Iran.

Abstract: Speech technology is one of the key technical issues involved in Information Technology as it constitutes an important aspect of Human Computer Interaction. Prediction of speech signal has applications in speech technology, especially in coding. Conventionally linear prediction is used. However, non-linear phenomena exist in speech production. Therefore, considering this non-linearity should lead to lower signal dynamics during coding with a consequent reduction in bit-rate and the needed bandwidth. This is studied in this paper using Feed Forward and Recurrent Neural Nets. It is shown through different evaluation schemes that the speech non-linearity is negligible and that non-linear speech prediction does not lead to an appreciable further reduction in the residual signal to be coded.

1. Introduction

The prediction of speech has applications in speech technology i.e. speech recognition, synthesis and coding. Linear prediction is used conventionally to reduce the redundancy of speech signal and decrease the bit-rate in coding. The reduction in bit-rate is achieved by coding the residual signal i.e. what remains from the speech once its predictable part has been removed. However, it is known that radiation effects from the lips and turbulences of the air flow from the lungs cause non-linear phenomena in speech production [1]. Therefore, considering the non-linearity in speech prediction is believed to result in lower dynamics of the residual signal to be coded. The non-linear prediction of speech can be achieved using Artificial Neural Nets.

2. Artificial neural nets and non-linear prediction.

Neural nets have been used extensively in non-linear problems for which an optimum explicit solution can not be found; among them non-linear prediction [2]. One reason for the popularity of the neural nets is the mere fact that they can automatically generate an optimum solution if it exists. The second one is the high speed of execution. Maybe finding the solution is time consuming, during training, but applying it, specially using parallel processing, is quite fast [3].

Hundreds of structures have been proposed for neural nets [4],[5]. However, from the structural point of view, these structures can be divided into two main groups: Feed Forward Neural Nets (FFNN) and Recurrent Neural Nets (RNN). In FFNNs, the mapping between the input and output remains unchanged once the training is completed and the output is calculated given the input regardless of the preceding and following states of the network; in other words the network is stationary. On the other hand, RNNs and among them Hidden Markov Models (HMM) and Hopfield networks are non-linear dynamic nets because of the presence of cyclic connections with their own complexity.

In most problems it is convenient to use multi-layer FFNNs (MLFN), due to their simplicity, when supervised learning is possible. In our study, MLFNs both with and without cyclic connections have been used. This choice was influenced by the fact that MLFNs can be used with and without direct connections from the first to the last layer [2], so permitting linear and non-linear Auto Regression (AR) modeling whilst the same structure with cyclic connections can be viewed as non-linear Auto Regression Moving Average (ARMA) modeling [6].

2. Non-linear prediction of the speech residual signal.

It is usually suggested that all linear trends of the input be removed before it is applied to a neural net. So doing will reduce the computation load leaving the neural net to extract the non-linear correlation left in the input. This can be done explicitly by calculating the linear prediction coefficients and the excitation signal and using this signal as the input of an MLFN or, implicitly by loading the calculated values as the fixed weights of the direct connections between the first layer neurons and the single neuron of the last layer of the network, as shown below, where the speech waveform is input to the structure arranged as Time Delay MLFN (TD-MLFN).

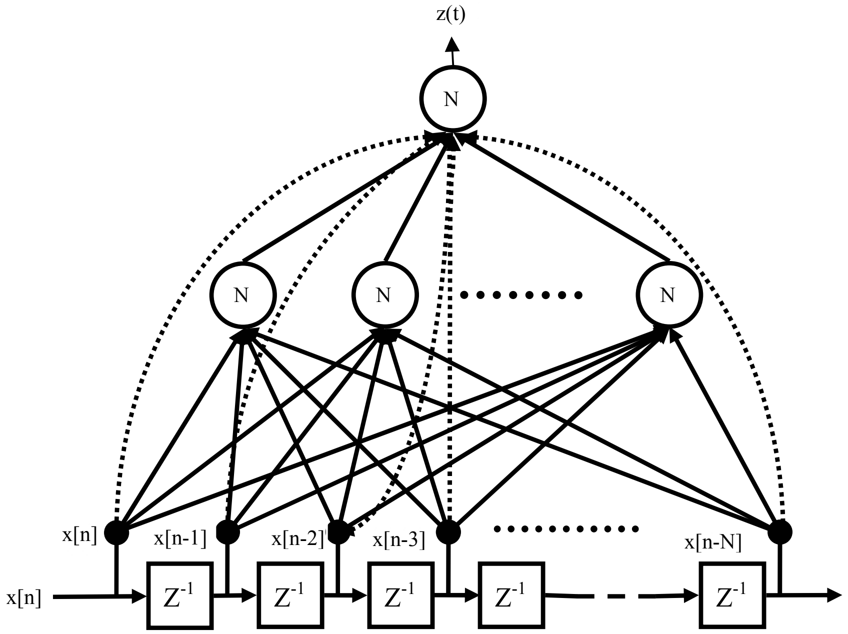

Figure 1: TD-MLFN with direct connections between the first and last layer.

Alternatively, instead of loading the pre-calculated linear prediction coefficients as weights of the connections shown above as dashed lines, the net can be let to calculate them in combination with all other weights. This is linear and non-linear prediction combined. When the network is to carry both linear and non-linear prediction, the neuron in the last layer must have a linear characteristic instead of the usual Sigmoid or Tansigmoid function.

3. The speech prediction using the TD-MLFN structure.

Generally speaking, the input of a neural net is a spatial pattern i.e. a vector of values. However, in some applications such as in speech processing, the input pattern is composed of values with temporal continuity. A conventional method for transforming this temporal signal into an input vector is to use a tapped delay line as shown in figure 1. When the prediction of the input signal is the issue, the output of the TD-MLFN is z(t)=x(n+1) where the network carries the prediction of the input one sample interval ahead. When the desired output is x(n+p), the prediction is for the pth interval ahead in time.

4. The back propagation training algorithm and its generalization.

The well known BP algorithm used for training of FFNNs can be generalized for training of MLFNs that include connections between a layer and any layer above it. This is shown as an example in [4]. The generalization of the BP algorithm to recurrent networks is given in [7]. This generalization as outlined in [4] has been used in this work. Other recurrent algorithms exist and can be found in [6].

The BP algorithm is the basis of training algorithm used in MATLAB NN Toolbox whose routines have been extensively used in our work. Nevertheless, we developed our own package of routines in Visual C++ using this algorithm and its generalizations mainly, to overcome the limitations of recurrent routines in MATLAB that are rather approximations.

4.1 The selection of the best training algorithm.

Many different variants of the BP algorithm exist for the training of simple MLFNs. Choosing the best algorithm is difficult and depends on many factors such as the network complexity, the number of samples in the training set, the initializations of weights and biases etc. MATLAB suggests the Levenberg-Marquardt (LM) algorithm as the best for networks with less than 100 eights [8]. This algorithm has been used here in batch training mode when using MATLAB Toolbox. The incremental training mode was not used because of its low speed. Batch Gradient Descent with Momentum was the method used for training in our developed package. In all cases the mean square error has been used as the optimization criterion.

4.2 Initialization.

When using MATLAB, the Ngugen-Widrow algorithm [ ] has been used for initialization. This algorithm is known to initialize the weights and biases so that fewer neurons are left inactive in the network and all neurons participate efficiently, increasing the speed of processing. The random initialization has been used in our package for simplicity.

4.3 Overtraining and the ways to avoid it.

One of the problems encountered during training is over-training or over-fitting. In this condition, the error in the training set is very low but when new data are employed the output error increases. In this situation the network is not generalized for new inputs. One of the methods used to avoid this problem is early stopping.

4.3.1 The early stopping of the training phase.

In this method the available data is divided in three sets: Training, validation and test. The training set is used for calculating weights and gradients during training. The validation set is also used during training and the error is calculated for this set although it is not used for network calculation. In normal conditions, continuing the training will reduce both the training set and the validation set errors but, when over-fitting is occurred the error for the validation set starts to grow. That is when training should be stopped. Naturally, the test set is only to assess the performance of the network once it is trained.

Early stopping has been used in this work.

5. Post-regression analysis of results.

The performance of a trained network can be assessed considering the errors corresponding to the above three sets. However, a regression analysis can be applied between the actual and the desired outputs. A linear regression is sought and the analysis is carried out calculating three parameters; namely m,b,r. The parameters m and b are respectively the slope and the distance from the zero origin in the best linear regression if there is any. If the network performs ideally m=1 and b=0. The parameter r is the correlation coefficient between the desired and actual output of the net. If r=1 then there is a complete correlation between the two.

6. The data-base.

The waveforms of phrases and words uttered by two male speakers were recorded at 11 and 22 KHz sampling frequencies and digitized with 8 and 16 bits. Then words were segmented into syllables to be saved in separate files as items of our data-base. The phonetic description of the files' contents and other characteristics such as the speaker code and the code of microphone used were attached to each file. A search engine permits to extract all files with a specific phonetic content and other needed characteristics such as the sampling frequency or bit representation for different experiments.

7. The experimental results.

The results obtained using TD-MLFNs are first reported. It is therefore assumed that this is the network used unless otherwise specified.

Since the initialization of network parameters was random it was necessary to repeat each experiment many times to ensure that local optimizations were avoided and a correct solution was achieved. When assessing the performance of the net on different inputs, such as in generalization, each experiment was repeated ten times and the best result saved for later comparison with similar experiments with other inputs.

7.1 Determining the network structure and dimension.

One of the important issues in neural computing is the selection of appropriate network structure and dimension. This problem has been dealt with in the literature. The following points are used usually as guidelines [3].

1- Use only one hidden layer. There are very few cases where using more than one hidden layer leads to better results. More than two hidden layers is not justified theoretically.
2- Use as few as possible neurons in different layers. Start with very few e.g. 3 and increase the number if necessary.

In some neural structures like TD-MLFNs, used for processing of speech signals, the correlation between input samples and other information such as the sampling frequency can be used for determining the minimum number of neurons needed in the first layer called also the sensors layer.

As for the speech signal sampled at almost twice the maximum frequency (i.e. 8 to 10 KHz), it is well known that considering neighboring samples more than 10 samples distant does not produce further reduction in the linear prediction error suggesting that the correlation drops to negligible value after 9 to 10 samples [9]. Therefore, from the point of view of the net's dimension, the 9-3-1 structure seemed appropriate to start with.

7.1.1 The optimum structure.

For input speech sampled at 11 KHz increasing the number of neurons in the hidden layer from 3 to 4 resulted in reducing the output error in most cases. But, for speech sampled at 22 KHz this reduction was insignificant. Increasing the number of neurons in the input layer from 9 to 15 reduced the output error for 11 KHz speech but again, this reduction was much less in the case of speech sampled at 22 KHz. It was concluded that the 9-3-1 structure was appropriate especially for speech inputs sampled at 22 KHz sampling frequency.

As for the neuron type, specified by the neuron's excitation function, changing it from tansigmoid (tansig) to pure linear (purelin) was without effect on the output. Therefore, it was assumed that the network for combined linear- non-linear prediction, with linear function for the output neuron, was structurally no different from the non-linear prediction network making the comparison easier.

7.2 The effect of bit resolution.

The results obtained on 22 KHz speech files at 8 and 16 bit resolutions showed that the training errors, for an equal number of training epochs, were very close and the network performed almost equally for either resolution. This showed the relative immunity of the network to quantization noise, a well known property, at least for high sampling frequency.

7.3 The effect of sampling frequency.

Different network structures and inputs with both resolutions were used, trained for the same number of epochs, with inputs sampled at 11 and 22 KHz. It was observed that a lower output error could be achieved for inputs sampled at 22 KHz as compared with those sampled at 11 KHz. This was interpreted as neural nets being capable of using more information presented in a wider input bandwidth.

7.4 Comparison of linear and non-linear prediction of speech.

Despite it was hoped that non-linear prediction being more general should result in lower output error, the error using a neural net and the linear prediction error calculated in a conventional manner were almost equal in all studied cases. This suggested that the redundancy in speech signal was mostly linear and non-linear prediction did not lead to an appreciable reduction in prediction error. This observation can be seen in the figure below.

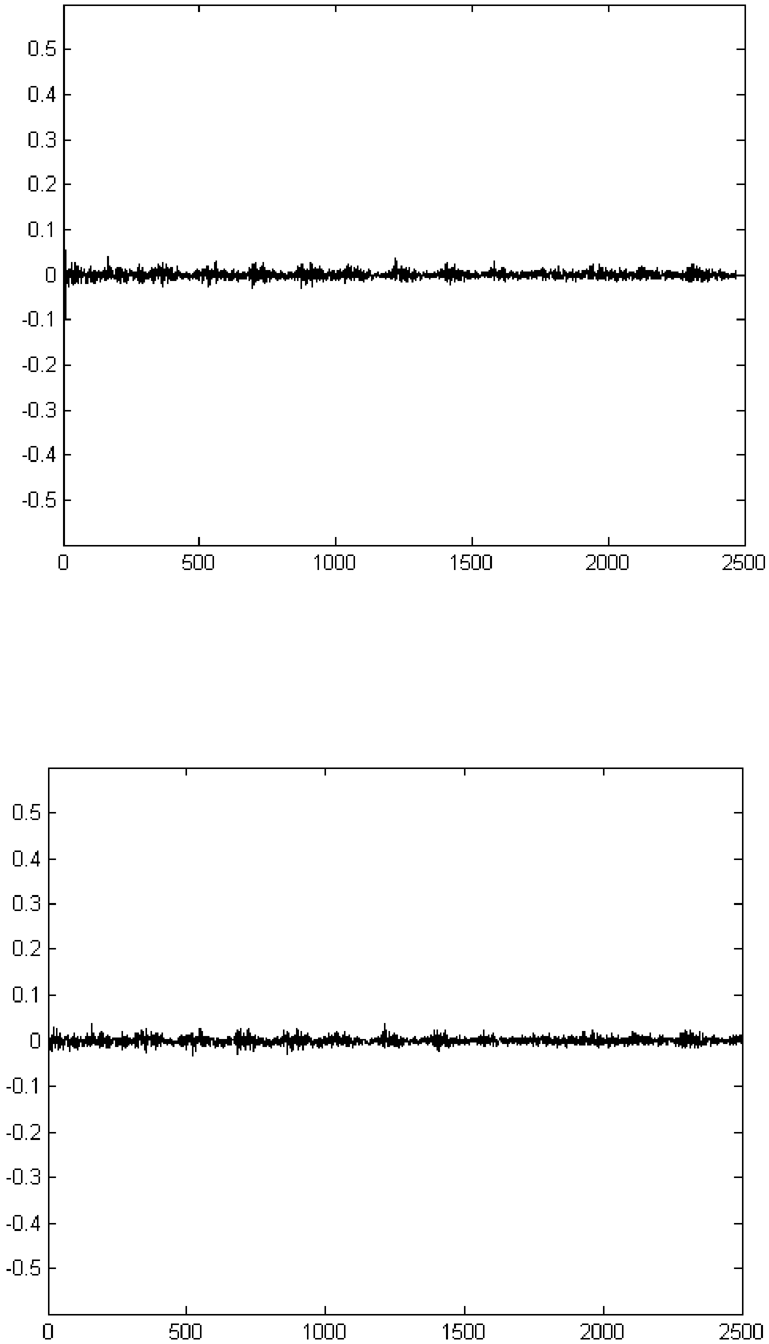

Figure 2: Prediction error signals; linear (top) and non-linear (bottom).

7.4.1 The non-linear prediction of the speech excitation signal.

When the linear prediction error signal, called also the excitation signal, was used as input the network was trained in the first few epochs showing that no further training was possible. In all cases the output error was only slightly smaller than the excitation input where the peaks were attenuated to some degree. Results obtained in these experiments confirmed what was mentioned earlier that the redundancy in speech signal was of a linear nature. This observation was confirmed by the post regression analysis as shown in figure 3.

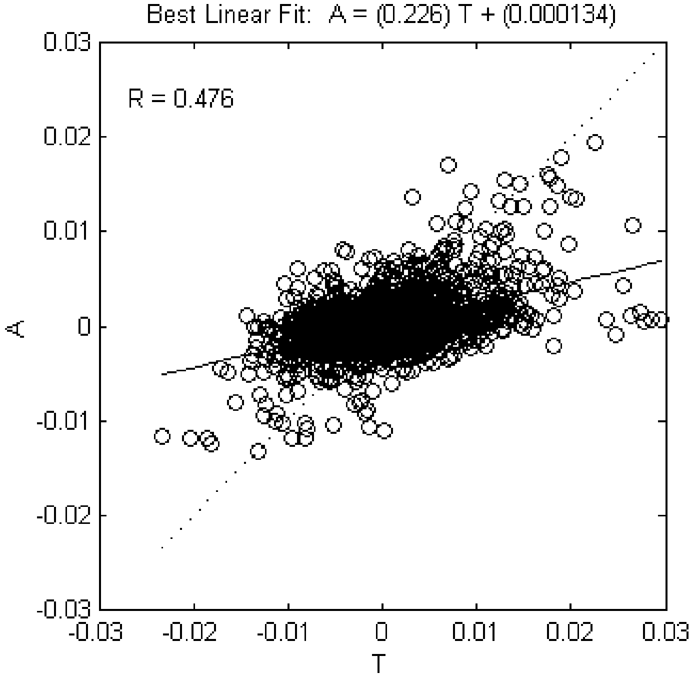

Figure 3: Post-regression analysis of non-linear prediction of speech excitation signal.

7.4.2 The combined linear – non-linear prediction of speech signal.

The combined linear and non-linear prediction using direct connections from the input layer to the output neuron with purelin excitation function led to a slightly lower error than the non-linear prediction using neural net. The difference being almost the same as in applying the speech excitation signal as input.

7.5 The generalization of the net.

7.5.1 The generalization for the same vowel.

The generalization was carried out first by training a net, for a given vowel, on the first of a sequence of files with the same content and using the trained net as the starting point for continuing the training with other files in the sequence. The procedure was repeated several times by changing the order of appearance of files in the sequence. The generalization depended on the set output error. For a moderate level of error the generalization was good even for a mixture of files from different speakers, resolutions and sampling frequencies. But, for low output errors the network did not generalize for 11 KHz speech inputs. Excluding these files permitted good generalization on 22 KHz speech files.

This observation was confirmed when using the early stopping of training as a means of controlling the generalization process. It can then be concluded that the generalization for the same vowel is quite good for inputs with 22 KHz sampling frequency. The bit resolution did not have much effect on this generalization.

7.5.2 The generalization for two or more vowels.

The training procedure was the same as above and two sets of experiments were conducted: One using a sequence of different inputs and using a trained net as the starting point for the next file and second using the early stopping for controlling the generalization. The result was almost the same. The network did not generalize well when 11 KHz files were included; but generalization was good with 22 KHz speech files.

7.6 Results with recurrent networks.

The experiments reported above were conducted with TD-MLFN in MATLAB. Nevertheless, they were confirmed using our own developed package. As for recurrent networks, the problem with MATLAB is that all recurrent connections include one sample delay and there is no obvious way avoiding these delays if a true instantaneous interaction is required. In our developed algorithm loops with and without delays were envisaged. Results using MATLAB recurrent nets were first confirmed employing our algorithm and then extended to true recurrent nets.

There was no difference in the above results when simple TD-MLFNs were replaced with recurrent substitutes where hidden layer neurons were connected two by two together. However, it is important to note that recurrent nets were trained, for the same output error goal, in less number of epochs; but because of the nature of the algorithm, the training took much longer.

8. Conclusion.

The ensemble of experiments conducted in this work shows that the redundancy in speech is of a linear nature and its non-linearity is not significant and does not warrant non-linear prediction especially when the linear prediction is conducted pitch synchronously where the prediction parameters are calculated for a short pitch period. The only advantage of non-linear prediction of speech using neural nets is perhaps its generalization power which can be achieved only if the input bandwidth is not limited in the sampling process.

REFERENCES.

1. H.M. Teager; “Some observations on oral air flow vocalization”; IEEE Trans. ASSP, Vol.28 (5), PP599-601.

2. A.S. Weigend; “Time series analysis and prediction”; www.cs.colorado.edu /~andreas /home.html.

3. T. Masters; “Signal and image processing with neural networks”, John Wiley & Sons, 1994.

4. N.K. Bose & P. Liang; “Neural network fundamentals with graphs, algorithms and applications”; Mc Graw Hill, 1996.
5. Limin Fu; “Neural network in computer intelligence”; Mc Graw Hill, 1994.
6. D.P. Mandic & J.A. Chambers; “Recurrent neural networks for prediction: Learning algorithms, architectures and stability”; John Wiley & Sons, 2001.
7. F.J. Pineda; “Generalization of back propagation to recurrent neural networks”; Physics Review Letters, 59, PP2229-2232.
8. MATLAB NN Toolbox – User’s guide; The MATH WORKS INC; www.mathworks.com.
9. S. Haykin & S. Kesler; “Prediction error filtering and maximum entropy spectral estimation”; in non-linear methods of spectral analysis, Springer-Verlag, 1983.