Turkish Journal of Electrical Engineering and Computer Sciences

Volume 9 | Number 2 Article 4

1-1-2001

Accurate Parameter Estimation for an Articulatory Speech Synthesizer with an Improved Neural Network Mapping

HALİS ALTUN

TANKUT YALÇINÖZ

K. MERVYN CURTIS

Turk J Elec Engin, VOL.9, NO.2 2001, © TÜBİTAK

Accurate Parameter Estimation for an Articulatory Speech Synthesizer with an Improved Neural Network Mapping

Halis ALTUN, Tankut YALÇINÖZ
Department of Electrical & Electronic Engineering,
Niğde University, Niğde-TURKEY
K. Mervyn CURTIS
School of Engineering, University of Technology, Jamaica and
School of Physics and Computing, University of West Indies-BARBADOS

Abstract

Neural network (NN) applications have recently been employed to extract the parameters of an articulatory speech synthesizer from a given speech signal. Results from these attempts showed that a single NN is insufficient to cover all of the possible configurations uniquely. Moreover, apart from their computational advantages, NN mapping is so far not superior to the other mapping techniques [1]. Thus there is a clear need to improve NN solution to the inverse problem.

Results from our earlier experiments with an articulatory speech synthesizer have shown that the statistical characteristic of the articulatory target pattern vectors can be exploited for an improvement in the estimation performance of a Multi-Layer Perceptron (MLP) NN [2]. In this paper, the effect of the modification to the distribution characteristic of the acoustic input pattern vectors will be investigated. The theoretical background for the effect of the input distribution characteristics on neural learning has been detailed elsewhere [3]. Empirical results for a more correct estimation of articulatory speech synthesizer parameters through exploiting the behavior of the Back Propagation (BP) algorithm are focused on here.

1. Introduction

In speech synthesis, there is a consensus among researchers that the articulatory speech synthesizer has the potential to be the ultimate solution to the synthesis of natural sounding, intelligible speech. It promises greater naturalness and allows for a greater flexibility in adjusting to the individual speaker [1,4,5]. Although remarkable attempts have been made towards this end, the problem of estimating control parameters for an articulatory synthesizer, from a given speech signal, still remains unresolved [1]. Due to its complex and ill-posed character, the inverse problem in the acoustic-to-articulatory mapping is a suitable application for neural network (NN) mapping. Algorithms for the acoustic-to-articulatory mapping using artificial NNs have recently been proposed for the extraction of the necessary parameters from the speech signal [6-8]. However, results from these attempts showed that a single NN is insufficient to cover all of the possible articulatory configurations uniquely. Moreover, apart from its computational advantages, NN mapping has not so far

proved to be superior to the other mapping techniques in the acoustic-to-articulatory inversion problem [1]. This makes it a necessity to improve the NN solution for the acoustic-to-articulatory mapping.

Attempts to improve the efficiency of NN computing have been reported. In the proposed solutions, the idea was either to enhance the BP algorithm itself [9], or to optimize the parameters of the algorithm such as learning rate [10], weights [11] and momentum term [12].

Here, a different method to obtain an improved neural learning will be demonstrated for the articulatory parameter estimation through modifying the distribution characteristics of the acoustic input pattern vectors according to the optimum statistical values stated in our earlier study [3].

Inversion in speech science has been understood as inferring the characteristic of the source or of the parameters of the filter, which is determined by the vocal tract. Within this paper, the inversion from the speech signal is conceptualized as obtaining the vocal tract area function, which is used as a control parameter in an articulatory synthesizer.

2. Inversion of the Articulatory Parameter

From a mathematical point of view, the inversion problem is classified as an ill-posed problem since the existence of a unique solution is not guaranteed. Also the inverse problem, in our case, demands knowledge about the mechanics of acoustic and articulation control processes of speech production. Mathematical analysis of conditions shows that a unique solution is not possible unless some values such as the length of the vocal tract, the boundary conditions, etc. are known. But, due to the absence of suitable automatic procedures for extracting such parameters immediately from the speech signal, inversion remains a very difficult problem [13].

In order to avoid such requirements, recent years have seen the increasing use of the mapping technique to ease such difficulties of the analytic model. One successful method is to use a codebook in which articulatory parameters and corresponding acoustic parameters have been paired to build up an entry [23]. The codebook is generated through applying some constraints on the vocal tract shape and spans the entire articulatory domain. The disadvantage of the codebook look-up method is that a small number of vectors in the look-up table can prevent one finding the global optimum. On the other hand, a large codebook, which is necessary to achieve good quality speech, demands a computational load.

Algorithms for acoustic-to-articulatory mapping using the artificial neural networks have recently been proposed [24-25]. Initial attempts trained a single NN to perform mapping from acoustic parameters, such as Cepstral or LPC coefficients, to articulatory variables. Results from these attempts showed that a single NN is insufficient to cover all of the possible configurations uniquely. Moreover, apart from its computational advantages, so far NN mapping is not superior to the other mapping techniques [1].

3. Improvement in Acoustic-to-Articulatory Inversion

Improvement in acoustic-to-articulatory inversion will be achieved through an improvement in neural learning. To this end, neural learning will be improved by creating a training data set which ensures a stronger correlation between the acoustic and articulatory domain vectors. Also, preprocessing of the acoustic input vectors will be employed in order to exploit the statistical nature of the acoustic input patterns, according to the results in [3].

3.1. Obtaining Training Pattern Vectors

The training set vectors have been created using a simplified, non-realistic Kelly-Lochbaum vocal tract (VT) model [32]. The optimized area functions of 10 English vowels are chosen from a set [26]. The assumptions made to simplify the VT implementation are as follows: VT consists of lossless uniform, concatenated acoustic tubes; the VT consists of a rigid wall; and the planar wave propagation is valid.

Then a linear interpolation is applied so that the population of the area functions is increased to 164, thus forming a larger training set. Acoustic input pattern vectors $\mathbf{x}_i$ are derived from the transfer function of the VT, which has been simulated in the MATCAD software package. The radiation load is approximated by a first order IIR filter, setting the reflection coefficient at the boundary of the last section as 0.99 to ensure IIR filter stability. The glottal impedance is neglected through setting the reflection coefficient at the first boundary to unity. Two examples from the training set are shown in Figure 1 and the corresponding articulatory and acoustic pattern values given in Table 1.

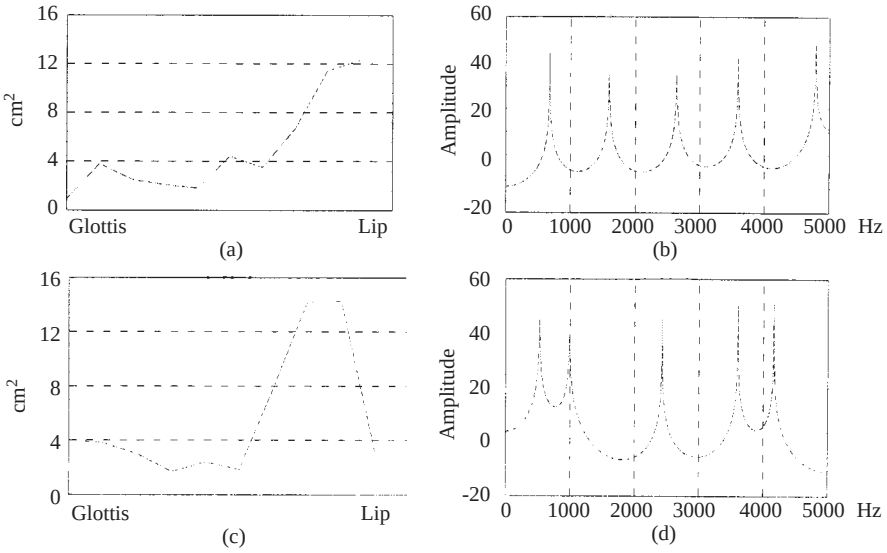

Figure 1. The VT area function and corresponding transfer functions: (a) and (b) for vowel /ae/, (c) and (d) for vowel /ao/

Table 1. Articulatory and corresponding acoustic vector values for /ae/ and /ao/

	Area function (cm^2)										Formant Frequencies (Hz)				
/ae/	1.7	7.2	1.65	1.52	2.45	3.3	6.7	7.5	7.4	3.7	686	1608	2611	3541	4776
/ao/	4	3.8	3	1.7	2.4	1.85	7.8	14.2	14.25	3.1	526	991	2424	3604	4163

3.2. Choosing Correct Training Patterns

In order to carry out acoustic-to-articulatory mapping successfully, the training data must have a strong correlation, as irrelevant data prevents NN from learning the correlation quickly [27]. Also, since inversion is an ill-posed problem, the acoustic data should be extracted as correctly as possible [28] and have strong correlation to the articulatory data. It has been shown that formant frequencies, as acoustic information, give the best performance in speech recognition [29] and it seems that they are more suitable in the inversion problem, at least for vowels [30], than other acoustic representations such as LPC and Cepstrum parameters

[13]. This is because the resonant frequencies depend primarily upon the vocal tract [31], whilst the LPC and Cepstrum coefficients are derived from the parameters of the vocal tract resonance alone and may prove to be weakly sensitive to variations in the articulatory parameters [13]. Therefore, acoustic input patterns are chosen so that they consist of the resonant frequencies obtained directly from the impulse response of the vocal tract instead of using the LPC or Cepstrum parameters.

In order to distinguish the vocal tract shapes for similar sets of formant frequencies, it is necessary to use some additional acoustic information such as formant damping or relative amplitude [29]. In our work, the distinctiveness of the acoustic vector is enhanced through using the 4th and 5th formants in addition to the first three, and further enhancement will be obtained through using a modification to the acoustic input patterns.

3.3. Effect of the Number of Formants on Neural Learning

If the 4th and 5th formants are included in the acoustic input pattern vectors, then a more distinctive input pattern results and the correlation between the input and output pattern vectors improves. As a result, improved neural learning can be achieved. In order to show the effect of the number of formants, two experiments were performed.

A neural network with single hidden layer of 18 nodes is employed. The number of output layer nodes is 10 and the number of the input layer is determined through the number of the acoustic input as either 3 or 5 formant frequencies. Also, the network parameters such as learning rate and momentum term are maintained in all attempts as 0.1 and 0.2, respectively. The predefined error threshold is kept constant for all the attempts at 0.0015 MSE and all networks are allowed to carry out up to 10^5 iterations, which is the iteration threshold, unless the error threshold is met before reaching the iteration threshold.

Two experiments are carried out. In the first experiment, input pattern distribution characteristics are not modified, while in the second experiment, input pattern vectors are subject to a modification which transforms the statistical characteristics of the input pattern vectors according to the optimum values stated in [3], which will be explained in the next section. Figures 2 and 3 show the results. In both experiments, neural networks trained with 5 formants successfully converge. The number of iterations is 45650 and 5200, respectively, for the first and second experiments. Results show that despite the fact that the first three formants are adequate to distinguish the acoustic patterns, a huge increase in the speed of convergence can be achieved when the 4^{th} and 5^{th} formants are included in the acoustic input vectors.

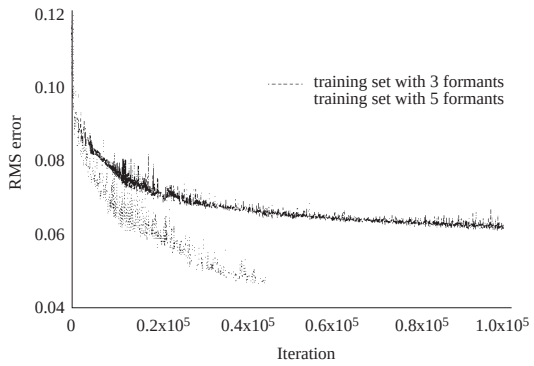

Figure 2. The effect of increasing the number of formants (first experiments)

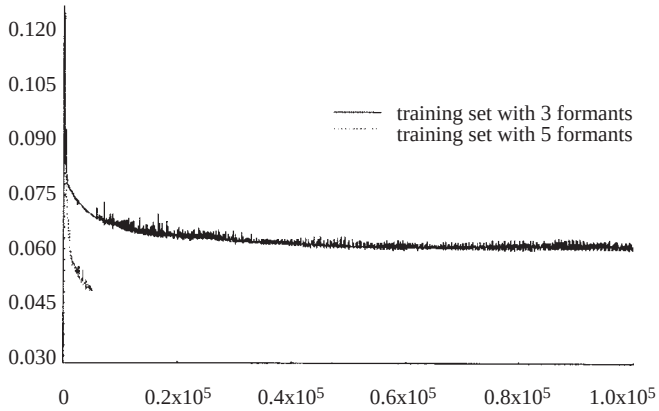

Figure 3. The effect of increasing the number of formants (second experiments)

3.4. Back-propagation algorithm

In order to investigate the effect of the statistical characteristics of the input pattern vectors on the MLP NN, let a NN with a single hidden layer have a vector space S. Let $\mathbf{x_i}$, $\mathbf{x_h}$ and $\mathbf{x_o}$, be activation vectors in this space, which present the node activation level of the layers. Assume that the input activation vectors $\mathbf{x_i}$ have a dimension of K, the hidden activation vectors $\mathbf{x_h}$ have a dimension of L and the output activation vectors $\mathbf{x_o}$ have a dimension of M.

$$\mathbf{x_i}^{(s)} = [x_1^{(s)}, x_2^{(s)}, \ldots, x_K^{(s)}]^T \tag{1}$$

$$\mathbf{x_h}^{(s)} = [x_1^{(s)}, x_2^{(s)}, \ldots, x_L^{(s)}]^T \tag{2}$$

$$\mathbf{x_o}^{(s)} = [x_1^{(s)}, x_2^{(s)}, \ldots, x_M^{(s)}]^T \tag{3}$$

where s is the number describing the individual training pattern with $s = 0, 1, ...n$.

The weighted connections between the input-hidden and hidden-output layers are $\mathbf{w_{ih}}$ and $\mathbf{w_{ho}}$. Training of the MLP NN using the BP algorithm requires a training set which consists of corresponding input and target pattern vectors, $\mathbf{x_i}$ and $\mathbf{t_o}$ respectively. Training continues until $\mathbf{w_{ih}}$ and $\mathbf{w_{oh}}$ are optimized so that a predefined error threshold is met between $\mathbf{x_o}$ and $\mathbf{t_o}$ as follows:

$$\mathbf{x_o}^{(s)} = \mathbf{t_o}^{(s)} \pm e \quad s = 0, 1, \ldots, n \tag{4}$$

where $\mathbf{e}$ is the predefined error tolerance and n is the number of patterns.

For the sake of clarity, let the input, hidden and output node activations, namely x_i, x_h and x_o, be termed "activation levels" rather than elements of activation vectors, $\mathbf{x_i}$, $\mathbf{x_h}$ and $\mathbf{w_o}$. Care should be taken that the activation levels of the hidden and output nodes, x_h and x_h, are determined by the algorithm itself (by the equations (9) and (10)) and it is not possible to modify their distribution characteristics directly. On the other hand, one can directly modify the activation level of the input node, x_i, and hence its distribution characteristics.

Using the new notation, interconnections between the nodes are adjusted by the amount of the weight update value as follows:

$$\Delta w_{ho} = -\eta x'_o \Delta x_o x_h \tag{5}$$

$$\Delta w_{ih} = -\eta x'_h \sum_{o}^{M} x'_o w_{ho} \Delta x_o x_i \tag{6}$$

$$\delta_o = x_o(1 - x_o)(t_o - x_o) \tag{7}$$

$$\delta_h = x_h(1 - x_h) \sum_{o}^{M} \delta_o w_{ho} \tag{8}$$

$$x_o = f_{\text{sig}} \left(\sum_{h} x_h w_{ho} \right) \tag{9}$$

$$x_h = f_{\text{sig}} \left(\sum_{i} x_i w_{ih} \right) \tag{10}$$

where $\Delta x_o = (t_o - x_o)$
$f_{\text{sig}}(\ )$: sigmoid activation function
δ : delta error term
η : learning rate
i, h, o : input, hidden and output layer indices
K, L, M : the number of input, hidden and output nodes, respectively
t_o : target value
x_o : output activation level
x'_o : derivative of the output activation level
x_h : hidden layer activation level
x'_h : derivative of the hidden layer activation level
x_i : actual input (input activation level)
w_{ho} : weights between hidden and output layer
Δw_{ho} : weight update for the hidden-output weights
w_{ih} : weights between input and hidden layer
Δw_{ih} : weight update for the input-hidden weights

The BP algorithm presented above has found widespread use in different areas. Thus there have been many proposed alterations to the algorithm to increase the speed of learning and improve the performance of the network. We propose a new method to improve the efficiency of learning through exploiting statistical characteristics of the acoustic input vectors.

The question we are looking for an answer to is how can the activation levels of the input, hidden and output layer be arranged so that strong weight update signals are produced by the BP algorithm? The result obtained from the analytical and statistical investigation of the above equations in [3] states that the optimum point for x_i is the upper bounds of activation domain [0,1], while it is 0.5, which is the middle of the activation domain, for x_o. However, there is a contradiction concerning the optimum value of the hidden layer node activation level, x_h. According to (5), a stronger weight update signal is produced for the hidden-output weights, w_{ho}, when x_h approaches the upper bounds. In contrast, the amount of the input-hidden weight update signal, Δw_{ih}, becomes very insignificant at this point according to (6). This compromise

creates a new optimum point for x_h. This optimum value for x_h depends on the number of input layer nodes K and that of the output layer nodes M. In Figure 4, the total weight update signal $\Delta w_{ih} + \Delta w_{ho}$ is given with respect to r_{node} , which is the ratio of the number of input and output nodes, which is calculated from the equations for an imaginary network of K-1-M structure, assuming optimal activation values for the input and output nodes. The figure shows the strength of the weight update signals versus hidden layer neuron activation for different ratios between input node K and output node M. As seen from the figure, the optimum expected value of x_h is shifted towards the middle of the activation domain where

$$r_{node} = \frac{K}{M} >> 1 \tag{11}$$

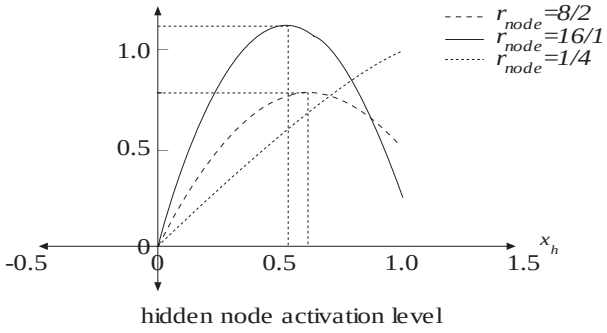

Figure 4. The weight update signals for the input-hidden and the hidden-output layer interconnections, w_{ih} and w_{ho}, versus hidden layer output signal x_h for different values of the input node/output node ratio, r_{node}

On the other hand, the optimum expected value of x_h will be around the upper limit of the domain if the ratio between the number of input and output nodes becomes very small where

$$r_{node} = \frac{K}{M} << 1 \tag{12}$$

The analytical analysis outlined above provides an isolated environment where activation levels are assumed to be independent in order to find out the optimum activation levels of the input, hidden and output nodes. Therefore, a statistical analysis should be performed in order to refine the assumption that the activation levels x_i , x_h and x_o are all independent.

Statistical investigation [3] shows that the probability of the hidden layer nodes x_h being in a non-strong production region, where the derivative of x_h is reduced more than 30% of its maximum, is decreased when the distribution characteristic of the acoustic input pattern vectors x_i shows a smaller expected value, $E[x_i]$, and standard deviation. However, the analytical inspection has shown that this type of distribution characteristic of the input layer activation x_i results in a small weight update signal for the input-hidden layer interconnections. This trade-off between the analytical and statistical findings on the optimum x_i implies that x_i should be transformed so that the expected value of the input pattern vectors, $E[x_i]$, should have a value around the middle of the domain, which is 0.5 when patterns are scaled within the range of [0,1].

4. Modifying the distribution characteristic of the acoustic input

Experiments are carried out using acoustic input patterns with different distribution characteristics. The first training set is created through maintaining the distribution characteristics of the original acoustic input patterns. The training set data has been scaled linearly between 0.05 and 0.95 before any further preprocessing. Then, the distribution characteristic of the acoustic input pattern vectors is investigated. Taking into account all individual input values, the expected value and the standard deviation of the acoustic input patterns calculated as 0.4827 and 0.2667, respectively. The distribution characteristic of the acoustic input patterns has a large standard deviation, which is a result of the scattered distribution of the acoustic input data as seen in Figure 5. This training set is called SET1.

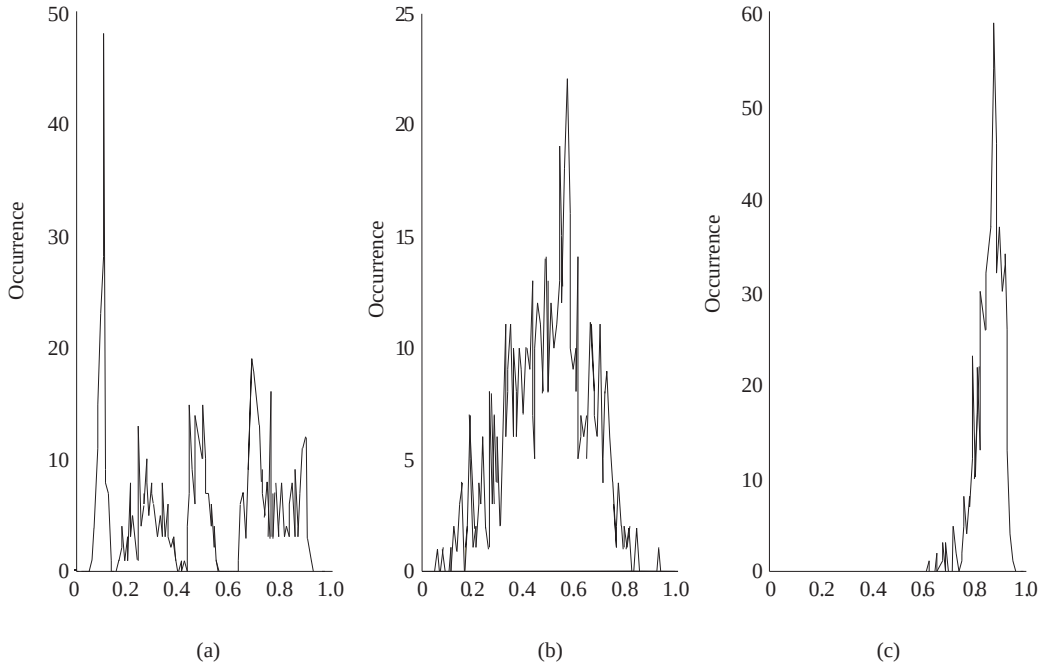

Figure 5. Distribution characteristic of the modified acoustic input pattern data a) SET1 b) SET2 c) SET3

The second training set is created using preprocessing which transforms the expectation value and standard deviation of the acoustic input pattern vectors into such values that are in the vicinity of the optimum expectation value and standard deviation. This training set is called SET2.

Another training data set is created in order to underline the effect of the distribution characteristic of the acoustic input pattern vectors. Hence, the preprocessing of the training set is deliberately arranged so that the expected value of the acoustic input data x_i diverges from the optimum expectation value towards the higher end of the activation domain. This training set is called SET3.

In Figure 5 and in Table 2, the distribution characteristics and the expected value and standard deviation for acoustic input data in the training sets are given.

Table 2. Statistical values of the acoustic input pattern data in SET1. SET2 and SET3

	SET1	SET2	SET3
Expected Value	0.4827	0.5028	0.8608
Standard Deviation	0.2667	0.1557	0.0511

4.1. Training NN Using Redistributed Acoustic Input Patterns

A two-layered NN with a structure of 5-18-10 is used. As the aim is to investigate the effect of the input data distribution rather than the optimization of the parameters, neural network parameters such as learning rate and momentum term etc. are heuristically set as 0.01 and 0.3, respectively. For the purpose of a fair comparison of the performance of MLP NN in each case, the initial state of the networks is kept identical for these experiments. This necessity is fulfilled using same initial condition for the input-hidden and hidden-output interconnections in each experiment. Uniformly distributed initial weight data, hence, is produced within the range of [-0.1, 0.1]. In addition, the predefined error threshold is kept constant for all the attempts at 0.0015 MSE and all networks are allowed to carry on up to the iteration threshold, unless the error threshold is met first.

The results from the experiments are given in Table 3 and the error curves are shown in Figure 6. From these results it can be seen that SET2 has a positive impact on the speed of the neural learning process. NN converges faster, by a factor of 8.13, when compared to SET1. On the other hand, a degradation in the speed of the learning process is observed, as expected, when NN is trained with SET3.

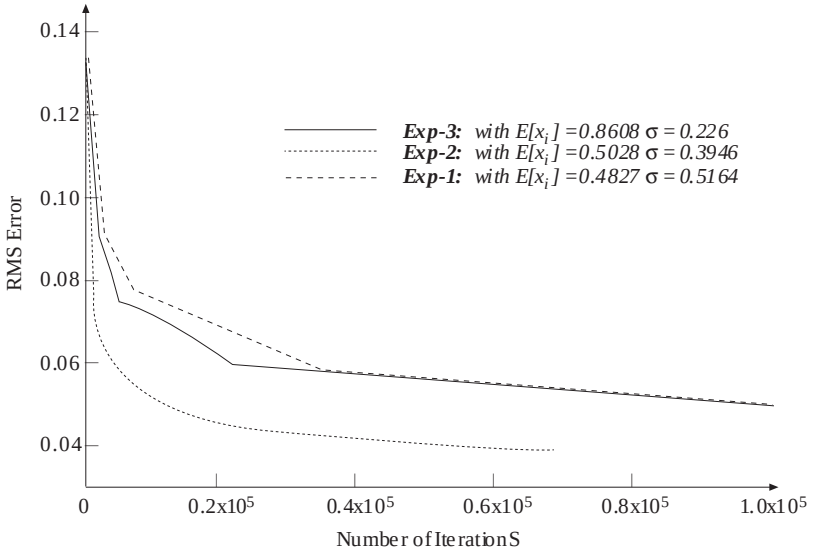

Figure 6. Error curves. Exp1: Trained with SET1; Exp2: Trained with SET2; Exp3: Trained with SET3

Table 3. The mse error and number of iterations required for each of the neural learning (Er_thr: Error Threshold: 0.0387 It_thr: Iteration Threshold: 100,000)

	SET1	SET2	SET3
Error	0.0497	Er_thr	0.0494
No of iterations	It_thr	68.500	It_thr

4.2. Input Distribution and Saturation in Hidden Layer

The effect of the acoustic input distribution characteristic can be also investigated in terms of degree of saturation in the hidden layer activation domain.

The distribution characteristic of the hidden layer activation levels x_h, which is taken at different intervals during the learning process, is shown in Figure 7. It is clear that the distribution characteristic of

x_h, which eventually affects the neural learning due to its direct effect on calculation of the weight update signals, is mainly dependent on the distribution characteristic of the input layer activation level x_i. It can be seen from the graphs that modifying the input layer activation levels x_i results in a change in the distribution of the hidden layer activation levels in the direction of the modification. If degree of saturation is defined as a function of the hidden layer activation level and its distribution characteristic, the improvement in neural learning can be calculated in terms of the degree of saturation of the hidden layer nodes x_h as follows:

$$\Theta(x_h, x'_h) = \frac{1}{\sum_h (n_h \cdot x_h \cdot x'_h) / \sum_h n_h} \tag{13}$$

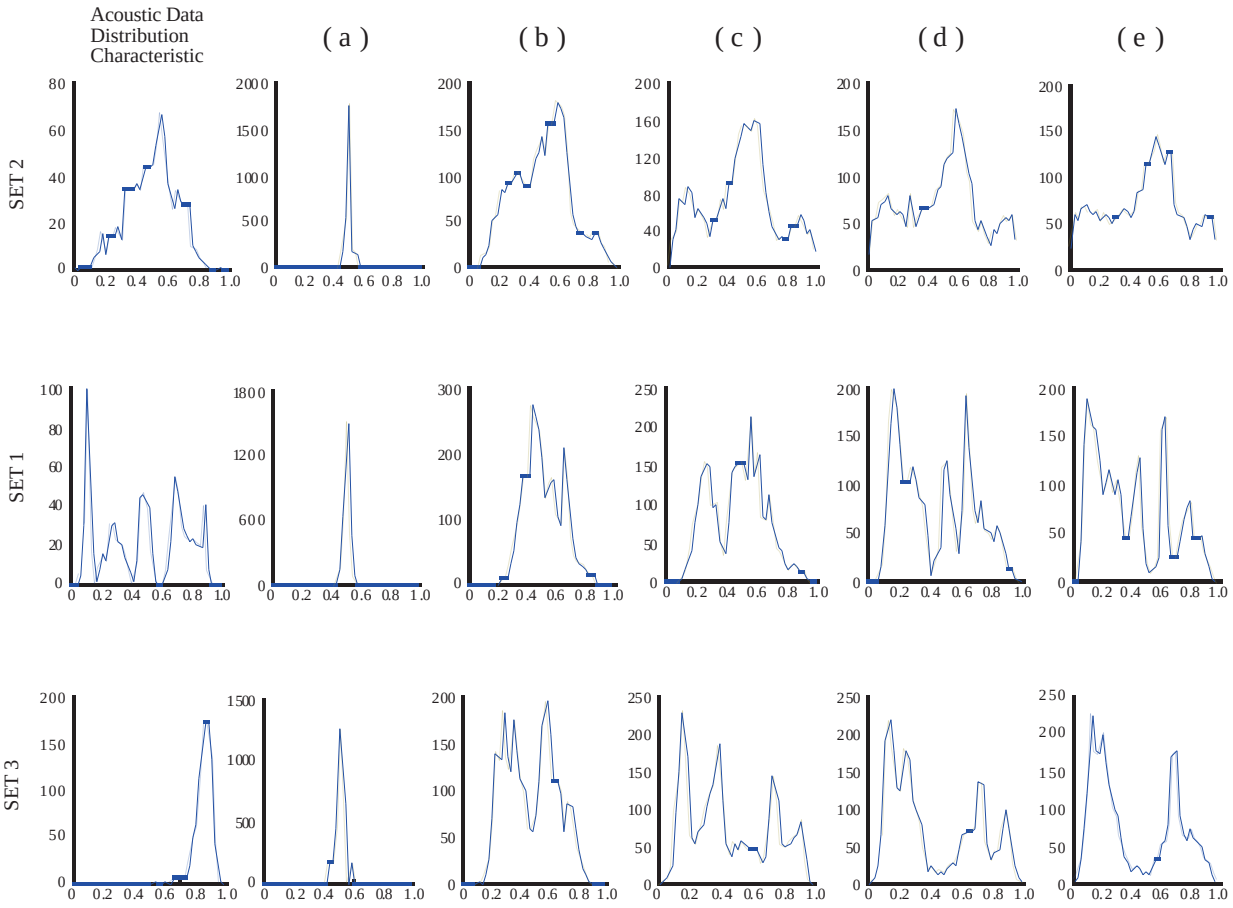

Figure 7. Evolution of the hidden layer activation characteristic for SET2, SET1 and SET3, respectively.

At the end of the learning phase, the degree of saturation in the hidden layer is reduced by a factor of 19.60%, a decrease from 13.082 to 10.517, for the NN trained with SET2. On the other hand, an increase in the degree of saturation is calculated as 2.0%, from 13.082 to 13.349, for the NN trained with SET3.

In Figure 7, it is also revealed that the transformation from the input layer to the hidden layer exhibits a linear-like nature. At the beginning of training, the distribution characteristics of the hidden layer activation show that none of the hidden layer nodes x_h are able to get any meaningful incoming signal from the input layer since the sum of products is near zero for each hidden node, leading to an activation level around the middle of the hidden layer activation domain (see Figure 7-a). As the learning process

continues, the weights are organized so that the sum of products for each of the hidden nodes x_h becomes slowly distinctive, diverging from its initial activation characteristic (Figure 7-b), with a more increasing consistency between the distribution characteristic of the input and hidden activation levels, x_i and x_h (Figure 7-c-d-e). This is particularly noticeable for SET1 and SET2. However, for SET3, the similarity between the distribution characteristic of the hidden layer activation and that of the input layer activation shows an inconsistency. This is an effect of the shifting acoustic input data towards the extreme end as a result of preprocessing. As acoustic data populate near the higher end, which is 1, the activation level of the hidden layer nodes is switched between the negative or positive saturation regions depending on the sign of the incoming signal to the hidden layer nodes. Investigation of weights during the learning shows that the symmetry and uniform distribution of the initial weights are lost, especially for w_{ih} . In the initial state, the weights w_{ih} and w_{ho} are distributed uniformly within a symmetric range of [-0.1, 0.1]. The range of the weights at different stages of the learning process for SET3 given in Table 4 reveals that the distortion in the symmetry is more prominent on the input-hidden layer connections w_{ih}, which increases the saturation probability in the hidden layer, while w_{ho} maintains its symmetric property. On the other hand, the distortion in the symmetry is not prominent for SET2 and SET1. The minimum and maximum range of w_{ih} at the end of the iteration are found to be [-17.503, 14.497] and [-8.865, 6.953], respectively, for SET1 and SET2. The distortion in the symmetry is calculated as 9.39%, 12.08% and 25.36% for SET1, SET2 and SET3, respectively.

Table 4. The range of the weights during evolution of the distribution of the hidden layer activation level (for SET3)

Input-Hidden Weights Range				Hidden-Output Weights Range		
Iterations	Min	Max		Iterations	Min	Max
1	-0.096	0.1		1	-0.013	0.098
20000	-4.507	7.529		20000	-3.921	4.873
40000	-5.225	12.709		40000	-4.348	5.289
60000	-6.925	14.761		60000	-4.599	5.643
80000	-7.771	15.54		80000	-4.967	5.891
100000	-9.468	15.902		100000	-5.16	5.896

4.3. Estimation Performance of the Networks

The generalization ability of the trained NNs are tested using 18 unseen acoustic-articulatory patterns. NNs trained with SET2, apart being the quickest neural learning, also yield a more correct estimation of the articulatory patterns. Total RMS error for these unseen articulatory patterns is calculated as 0.0761, 0.0752 and 0.0652 for the NN trained with SET3, SET1 and SET2, respectively. NN trained with SET2 exhibits a reduction in RMS of 13.29% in the estimation of the unseen articulatory parameters. Also, the RMS error between the original and constructed impulse spectra should be considered due to the ill-posed, one-to-many mapping between the acoustic and articulatory pattern vectors. The RMS error in the corresponding original and estimated impulse spectra are calculated as 6.102 and 5.304 for SET1 and SET2, respectively which is an improvement of 13.08% in the RMS reduction.

5. Conclusion

It is shown that, in estimating the articulatory control parameters of an articulatory speech synthesizer, an increase in the learning speed and in the accuracy of the estimation performance of an NN can be achieved

when the statistical characteristic of the acoustic input pattern vectors are statistically adjusted according to the optimum statistical values stated in [3]. This also results in a decrease in the degree of saturation of the hidden layer nodes. If the modification to the statistical characteristic of the acoustic data is not appropriate, it results in a slowing down in the learning process and a degradation in the estimation performance of the NN, as illustrated in the case of SET3. It is proved that an appropriate modification should be employed in order to enhance neural learning for a particular problem, incorporating the underlying feature of the problem in hand. As demonstrated above, a suitable modification to the acoustic input data improves the convergence rate, as in the case of SET2, by a factor of up to 8.78 when compared to SET1. The improvement in the estimation performance of the NN is also calculated. Total reduction in the RMS error of the estimated articulatory parameters and reconstructed acoustic patterns is calculated as 13.29% and 13.08%, respectively.

References

[1] J. Schroeter, M.M. Sondhi, "Techniques for Estimating Vocal-Tract Shapes from the Speech Signal.", IEEE Transactions On Speech And Audio Processing, 2, 133-149, 1994.

[2] H. Altun and K.M. Curtis, "Improving the estimation of the articulatory parameters for an articulatory synthesizer using an MLP neural network with vector scaling procedure", Proc. of 14th IEEE Int. Conf. on Electronics, Circuits, and Systems, ICECS'97, Cairo, 1, pp. 29-33, 1997

[3] H. Altun, K.M. Curtis, "Exploiting the statistical characteristic of the speech signal for an improved neural learning in MLP neural network", The 1998 IEEE Neural Networks for Signal Processing, NNSP'98, Cambridge, 1998

[4] D.C. Klatt, L.C. Klatt, "Review of Text-to-Speech Conversation for English." JASA, 82, 737-793, 1987.

[5] G. Frant, "What can basic research contribute to speech synthesis." J. Phonetics, 19, 75-90, 1991

[6] M.G. Rahim, C.C. Goodyear, W.B. Kleijn, J. Schroeter, M.M. Sondhi, "On the Use of Neural Networks in Articulatory Speech Synthesis." JASA, 93, pp.1109-1121, 1993.

[7] T. Kobayashi, M. Yagyu, K. Shiriai, "Application of Neural Networks to Articulatory Motion Estimation.", IEEE Trans. Acoust. Speech Signal Process, pp.1089-1100, 1991

[8] J. Zacks, R.T.Thomas, "A new neural network for articulatory speech recognition and its application to vowel identification." Computer Speech and Language, 8, pp.189-209, 1994

[9] S. Kodiyalam, R. Gurumoorthy, "Neural networks with modified backpropagation learning applied to structural optimisation", AIAA Journal, 34, pp. 408-412, 1996

[10] H.B. Kim, S.H. Jung, T.G. Kim, K.H. Park, "Fast learning-method for backpropagation neural-network by evolutionary adaptation of learning rates", Neurocomputing, 1996, 11 (1), pp. 101-106

[11] J.S.N. Jean, J. Wang, "Weight smoothing to improve network generalisation," IEEE Transactions on Neural Networks, 5 (5), pp.752-763, 1994

[12] A. Kanda, S. Fujita and et al, "Acceleration by Prediction for Error Backpropagation Algorithm of Neural Networks", Systems and Computers in Japan, 25 (1), pp. 78-87, 1994

[13] V.N. Sorokin, "Determination of vocal-tract shape for vowels." Speech Communication 11, 71-85, 1992.

[14] D. Beautemps, P. Badin, R. Laboissire, "Deriving vocal-tract functions from midsagittal profiles and formant frequencies: A new model for vowels and fricative consonants based on experimental data." Speech Communication 16, 27-47, 1995.

[15] J.S. Perkell, "Physiology of speech production: results and implications of a quantitative cineradiographic study", MIT Press, 1969

[16] J. Schroeter, M.M. Sondhi, "Speech coding based on physiological models of speech production", in : S. Furui and MM Sondhi, Eds., Advances in Speech Signal Processing (Marcel Dekker, New York), 231-268, 1992

[17] M.G. Rahim, C.C. Goodyear, W.B. Kleijn, J. Schroeter, M.M. Sondhi, "On the Use of Neural Networks in Articulatory Speech Synthesis.", JASA, 93, 1109-1121, 1993

[18] G. Papcun, J. Hchberg, T.R. Thomas et al., "Inferring Articulation and Recognizing Gestures From Acoustic with A Neural Network Trained on X-ray Microbeam Data", .JASA 92(2), 688-700, 1992

[19] J.L. Kelly, C.C. Lochbaum, Speech Synthesis. Proc. Fourth Intern. Congr. Acout., Paper G42, 1-4., 1962

[20] M. Rahim, Artificial Neural Networks in Speech Analysis/Synthesis, Chapman & Hall, 1994

[21] N. Littestone, "Learning Quickly When Irrelevant Attributes Abound: A New Learning-threshold Algorithm", Proceedings of the 28th IEEE Conference on Foundations of Computer Science, 68-77, 1987

[22] V.N. Sorokin, A.V. Trushkin, "Articulatory-to-acoustic mapping for inverse problem." Speech Communication 19, 105-118, 1996.

[23] A. Soquet, M. Saerens, "Vowels classification based on acoustic and articulatory representations." ICPhS 3, 322-325, 1995.

[24] Q. Lin, G. Fant, "An Articulatory Speech Synthesizer Based on A Frequency-Domain Simulation of the Vocal Tract." IEEE 0-7803-053209/92, 1992

[25] L.R. Rabiner, R.W. Schafer, Digital Processing of Speech., Prentice & Hall. 1975.

Appendix: A.1.

Modification of the Distribution Characteristics: Scaling Functions

SET1 is created through scaling linearly all values between 0.05 and 0.95.

To create SET2 the acoustic domain is split up into five sub-regions through determining the lower and upper limits for each formant region according to the minimum and maximum values of the individual formants as seen in Table A.1.

The acoustic data is then scaled using a linear scaling function of the form

$$f(x) = \frac{(Y_2 - Y_1)X_1Y_2 + X_2Y_2}{X_2 - X_1}$$

where $Y_1 = 0.05$ and $Y_2 = 0.95$, X_1 and X_2 are the lower and upper limits of a sub-region given in the table. The overall effect of each linear scaling is equal to performing a non-linear scaling over the whole acoustic input domain.

SET3 is created employing a logarithm scaling function, which shifts the expected value toward the upper bound. The function is given as

$$f(x) = \frac{\log(x + 1.2 - X_1)}{\log(X_2 - X_1)}$$

where X_1 and X_2 are the lower and upper limits of a sub-region given in the table.

Table A.1 The range of defined sub-regions for each individual formants

	Minimum (X_1)	Maximum (X_2)
F1	200	800
F2	750	2400
F3	2000	3100
F4	3000	4100
F5	3800	5000