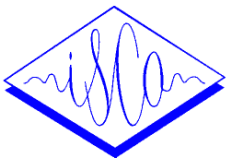

2nd European Conference on
Speech Communication and Technology
EUROSPEECH '91
Genova, Italy, September 24-26, 1991

Neural Network Classification of Complex-valued Speech Features

E.C. Andrews and J.S. Mason

Department of Electrical and Electronic Engineering
University College, SWANSEA, UK.

Abstract

Most speech features are inherently complex but usually only their magnitude is considered in terms of spectral distortion measures. DFT and cepstral spectra are typical examples of this, where the phase information is usually thought to be of little value and is therefore discarded. This paper describes a new form of neural network that is inherently complex. We propose its use in applications where the input to a pattern recognition task contains complex information, and choose the task of speaker verification.

The complex feature we consider here is the DFT of cepstral time series spanning a single utterance. In generating such features we show the effects of sampling rate and aliasing on the 2D mel-cepstra.

The role of the non-linearity in the complex network is of paramount importance. We propose functions suitable for this case, since the standard sigmoid is inappropriate.

To evaluate this new structure the task of speaker verification is chosen. Preliminary results are promising, supporting the case for the complex net.

10.21437/Eurospeech.1991-24

1 Introduction

Short term spectral estimates, in one form or another, underly most front-end feature extraction methods and are common to speech coding, speech recognition and speaker verification. A cepstral form is often chosen owing to its accurate modelling and convenient Euclidean spectral distance measure.

Although such estimates are inherently complex, it is usual to consider only the magnitude component, ignoring all phase information. This can be at least partly justified in that the relative phases within such spectra are known to be of secondary importance in the case of speech. However, the phase does encode inter-frame time sequence information and might in the right circumstances be useful in both coding and recognition tasks.

Our hypothesis here is that phase, represented by arithmetically complex features, might be useful; this in turn prompts the idea of an artificial neural net which is inherently complex. We consider a multilayer perceptron (MLP) with such a form. There are many possible levels of complexity within the net itself. For example it could take complex-valued inputs, have complex-valued weights, perform complex multiplications and additions, include complex-valued non-linearities and give, where appropriate, a complex-valued output and associated complex-valued error. This paper proposes and investigates such a structure with differing degrees of the complex components. In particular we focus on the choice of non-linearity.

The first logistic function considered comes directly from the standard sigmoid, extended to a complex form. However, this is shown to be inappropriate, and alternative functions are proposed that meet the restrictions imposed by back-propagation training, namely a continuous, differentiable function.

Finally we consider an application of the complex MLP (CMLP) to speaker verification, using a 2-D mel cepstra feature representation originally proposed by Kitamura [1].

2 Complex Nets

In this section the implications of introducing some form of complex modelling into an MLP are examined. The extent of complexity and the choice of non-linearity itself are the main subjects discussed.

2.1 Implications of Complexness

Clearly for a complex network to be useful the input signal must be complex. Complex signals may be classified using *standard* real-only networks by the simple procedure of providing a real and imaginary pair of vectors as input. Theoretically such an arrangement should be sufficient for any task, since an MLP of sufficient size can perform any arbitrary mapping from its input to output space[2]. However, the key here is training, and current knowledge is insufficient to give such a solution in practice. Thus the explicit inclusion of a complex representation within an MLP is investigated.

There are several degrees to which an MLP may be made complex.

1. As mentioned above, it may be sufficient solely to present the real and imaginary parts to a traditional network without reference to complex values. NB this has no explicit complex representation within the net. Some licence is needed to call this network complex as it has no complex components. However, it is included here to provide a starting point.
2. The inputs and first layer of weights may be complex, with phase information discarded between the summation and non-linearity in the first hidden layer. This would mean that standard sigmoids could be used for the non-linearity.
3. All non-linearities and weights between the inputs and outputs are complex, with phase information discarded before the final output.
4. Finally, a fully complex structure is one with every weight and non-linearity in the net complex, including the final output.

These different classes are shown in Figure 1. In applications where both input and output signals are complex, eg (non-linear) filtering, a fully complex structure of the fourth type is likely to be the most suitable. However, in a classification task such as speaker verification the MLP output is treated as a likelyhood measure, where for example higher values indicate a greater belief that the input corresponds to the specific class. Thus only a *real* value is appropriate at the output ruling out the 4th structure from consideration.

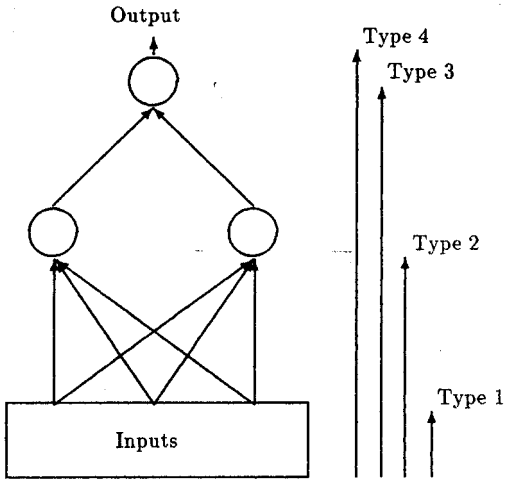

Figure 1: Levels of complexity within a net structure: Type 1 real and imaginary inputs; Type 2 as Type 1 but with complex weights; Type 3 as Type 2 but with complex non-linearity; Type 4, fully complex from input to output.

2.2 Non-linearities in complex networks

We now turn to the non-linearity to be used in a complex net. It should be noted that this choice is very important to the operation of the network and is a dominant factor in describing the shape of the error space.

The standard sigmoid function shown in Figure 2a is

$$f_1 = \frac{1}{1+e^{-z}}$$

Unfortunately, when z is complex this function is discontinuous at $j(1+2k\pi)$. Since the gradient descent optimisation procedure requires a monotonic error surface f_1 is clearly unsuitable for use in a complex network.

A simple extension to this standard sigmoid, placing separate sigmoids on the real and imaginary components of the signal is

$$f_2 = \frac{1}{1+e^{-\Re(z)}} + \frac{i}{1+e^{-\Im(z)}},$$

shown in Figure 2b. This function is both continuous and monotonic and therefore suitable for use in back-propagation. However, one limitation (which might or might not be important) is that this function makes no use of the relationship between real and imaginary parts, treating them as independent signals. Thus in this case it is only in the weighting of node inputs that the net is truly complex.

One function that does combine the real and imaginary parts is

$$f_3 = \frac{e^{i\angle z}}{1+e^{-\frac{\Re(z)+\Im(z)}{1+|z|}}}$$

which has a similar form to f_2 but is 'squashed' at points away from the real axis, Figure 2c.

The computational implications for the derivative of this function make it a subject for future work. Initially we consider f_2 which has a simple derivative and isolates complex effects to the weighting process, properties more suited to a preliminary investigation.

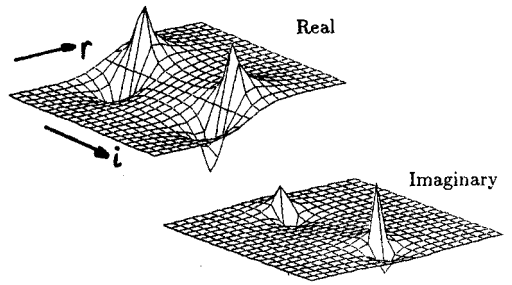

Figure 2a: Real and imaginary parts of the standard sigmoid $f_1(z) = \frac{1}{1+e^{-z}}$

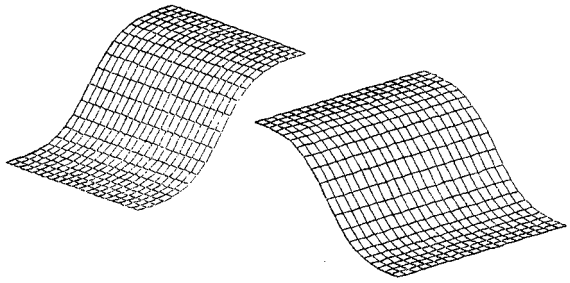

Figure 2b: Real and imaginary parts of a simple extension to the standard sigmoid, $f_2(z) = \frac{1}{1+e^{-\Re(z)}} + \frac{i}{1+e^{-\Im(z)}}$

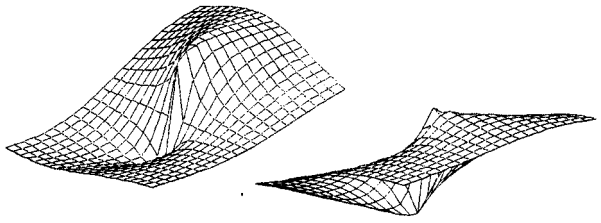

Figure 2c: Real and imaginary parts of the function $f_3(z) = \frac{e^{i\angle(z)}}{1+e^{-\frac{\Re(z)+\Im(z)}{1+|z|}}}$

3 2D Cepstra Features

This form of front-end feature was first proposed by Kitamura as early as 1976 [1]. Conceptually it is convenient to consider a series of standard cepstral features obtained by sliding the short term analysis window along the time course of the utterance. The dynamics of these features, treated as individual series, are often derived from regression analysis or simple differencing and used as secondary features in both speech and speaker recognition. Kitamura however proposed an interesting alternative approach, based on spectral analysis of these series, and has examined such features in tasks such as pitch prediction, speech recognition and speaker recognition [1][3][4]

3.1 Definition

A single window of speech $s(t)$ has the nth order cepstral representation defined as

$$c(p) = FT^{-1}(log||FT(s(t))||), \qquad \text{for } p = 1, 2, \ldots, n$$

Since the analysis window may occur at any point of the utterance it is useful to consider $c(p, w)$, the coefficient p of the cepstral vector representing the w'th window of speech $s(w, t)$,

$$\begin{aligned} c(p,w) &= FT^{-1}(log||FT(s(w,t))||) \\ &= FT^{-1}(log||F(w)||), \end{aligned}$$

where FT stands for the Fourier transform operation, and $F(w)$ is the Fourier transform of the w'th frame of speech.

Consider the n time series of cepstral coefficients obtained by sliding the analysis window along the time course of $s(t)$ is steps of ΔT samples. The 2D spectra, C_{2D}, consists of the n spectra, one for each cepstral series p, calculated over a number of windows N that we choose to have span the whole utterance. ie

$$\begin{aligned} C_{2D} &= C(p) & 0 < p \leq n, \\ \text{where } C(p) &= \text{spectrum}(c(p,w)) & 0 \leq w \leq N-1 \\ &= \frac{1}{N} \sum_{w=0}^{N-1} c(p,w).e^{-j2\pi \frac{w}{N}} \end{aligned}$$

3.2 Illustration of a typical 2D cepstra feature

To illustrate the 2D cepstral feature we take an example utterance of the word 'c', sampled originally at 10kHz. The original sampled waveform is shown in Figure 3, followed by the first 5 cepstral coefficients. Both mel and inverse variance weighting are applied.

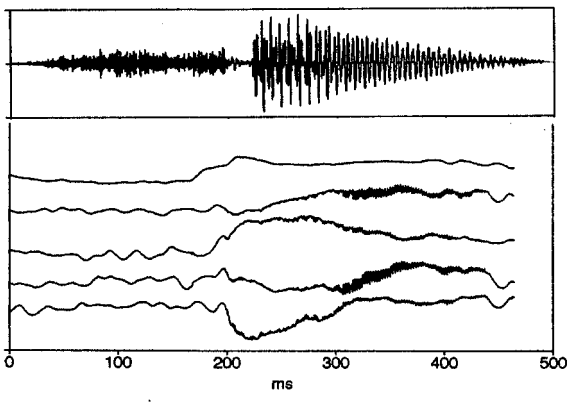

Figure 3: Cepstral time series for an utterance 'c'. High frequency components (about 500Hz) are seen on all but the first coefficient. These are due to dominant periodicities in the original speech beating with the analysis window.

For the purposes of this illustration the series are calculated using an overlap of 254 samples between adjacent 256 sample frames, corresponding to a frame rate of 0.2ms. Note that most of the coefficients have a superimposed noise towards the end of the word. This originates from a dominant periodicity in the original speech signal beating with the duration of the front-end Hamming window. It is also interesting that there are correlated areas between the coefficients, particularly at the middle and over the last section of the word.

Next consider the DFT-derived spectra of these time series, calculated over the whole utterance. The spectrum for a single cepstral coefficient is found to be very noisy and therefore the spectra shown here are averages calculated across two speakers and a number of utterances.

Figure 4a shows the spectra of the first 5 cepstral series. Since the cepstra are calculated at 0.2ms intervals the Nyquist frequency of these

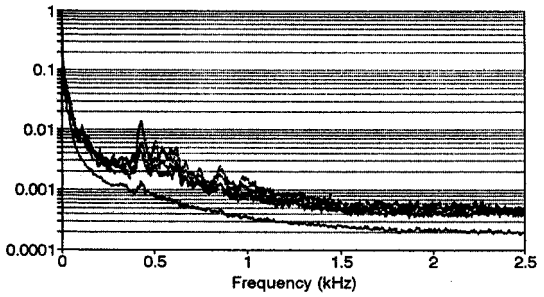

Figure 4a: Magnitude spectra of the first 5 cepstral series averaged over two speakers and several Eset utterances. The peaks at 500Hz are evident in the time waveforms in Figure 3

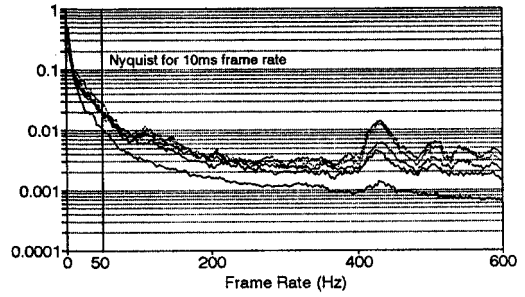

Figure 4b: Expansion of Figure 4a showing the first 600Hz. The Nyquist frequency corresponding to the common 10ms frame rate is included to show typical aliasing levels.

spectra is at 2.5KHz. This is a very fast rate compared with that normally used but is adopted here for illustrative purposes. Of course, it is preferable to calculate these series at a minimum rate commensurate with giving an accurate spectral estimate, ie without introducing excessive aliasing. From the graph it is clear that to avoid aliasing of the high frequency 'beats' mentioned above, and clearly evident in the time plots (Figure 3), it is necessary to work at a frame rate equivalent to 1ms or more, as the peak occurs in the region of 500Hz. A commonly used frame rate is 10ms, only $\frac{1}{10}$th this value, is indicated in Figure 4.

Of course the frequency and relative amplitude of these peaks are a function of the original speech. The ones shown here are not hand selected but are typical Eset examples. The question of whether such aliasing has any significant contribution in dynamic feature analysis is not investigated here, but is raised as an interesting question. For the example shown in Figure 3 the ratio of the magnitude of the 500Hz component to the low frequency peak is in the order of 5:1 for the worst cases, coefficients 2 and 4.

For our work we wish to examine the spectra in terms of the fall off at low frequencies and to determine the number of (complex) low frequency DFT components to be fed to the neural classifier.

In choosing the cepstral time series and then taking their DFT we are essentially performing integration over time. We postulate that this is acceptable and possibly beneficial in speaker recognition since the relative timing of events is likely to be less important than in for example speech recognition. In this sense we are examining an average characteristic over the utterance length, rather than its temporal detail, although with the inclusion of phase in the complex case a time related component is indirectly included.

Obviously the dominant information is at the lower frequencies since the vocal tract can change only at a relatively low rate. Figure 4b shows the lower quarter of the spectrum, corresponding to frequency components up to around 600Hz. Only the first quarter of these samples have values above the 'noise' level, allowing all but the first 64 samples to be discarded to leave a spectrum containing frequencies up to 160Hz in the 2D cepstra.

4 Speaker Verification Experiments

As a first stage in evaluating the utility of complex networks a study of a speaker verification task is proposed.

Our front-end features are 8th order mel cepstra. We acknowledge that higher order analysis would give better absolute performance but our objective here is to perform a relative comparison. The 2D spectrum is calculated at a frame rate of 0.4ms with components after the 64th discarded.

Initially we examine the 2D cepstra as magnitude spectra, the using real and imaginary pairs of coefficients in a type 2 complex net, followed by a type 3 structure which includes the complex non-linearity defined above and shown in Figure 2.

Training is performed for each true talker with nine imposters over a vocabulary of the alphabet. Ten different 'unseen' imposters are used for testing.

Initial results, as yet statistically insignificant, support our initial hypothesis showing consistent improvements with the inclusion of additional complexity.

5 Summary and Conclusions

This paper proposes the concept of a complex MLP for use in the classification of inherently complex features. The standard sigmoid commonly used as the non-linearity in MLP's is inappropriate in the case of a complex net and hence we propose alternatives which satisfy the conditions imposed by training procedures.

The complex feature we consider is a transform of the mel-cepstra time series spanning a single utterance. Here we show the importance of frame rate in terms of aliasing effects and show a relatively high noise level superimposed on the series due to frame beating with dominant periodicities in the original signal.

These complex features are deemed suitable for the task of speaker verification in that the DFT process integrates over the time course. This would make it less suitable for speech recognition where fine temporal detail is likely to be more important.

The main purpose of this work is to examine complex neural nets. Experiments are continuing but our preliminary results demonstrate the benefits of incorporating complex arithmetic into a standard MLP.

References

[1] T. Kitamura and S. Imai. Pitch Determination by Two-Dimensional Cepstrum. *Research Laboratory of Precision Machinery and Electronics, Tokyo Institute of Technology*, March 1976.

[2] R. P. Lippmann. An Introduction to Computing with Neutral Nets. *IEEE ASSP Magazine*, pages 4–22, April 1987.

[3] T. Kitamura and S. Imai. Speech Analysis Using Two-Dimensional Cepstrum. *Research Laboratory of Precision Machinery and Electronics, Tokyo Institute of Technology*, March 1977.

[4] T. Kitamura and S. Imai. Speech Synthesis Using Two-Dimensional Cepstrum. *Research Laboratory of Precision Machinery and Electronics, Tokyo Institute of Technology*, March 1977.