ISCA Archive
http://www.isca-speech.org/archive

6th European Conference on
Speech Communication and Technology
(EUROSPEECH'99)
Budapest, Hungary, September 5-9, 1999

Real-time Speech Modeling using Computationally Efficient Locally Recurrent Neural Networks (CERNs)

John J.Soraghan, Amir Hussain and Ivy Shim*

Signal Processing Division, University of Strathclyde, Glasgow G1 1XW, UK
email: j.soraghan@eee.strath.ac.uk
*Dept. of Electronic Engineering and Physics, University of Paisley, Paisley PA1 2BE, UK

10.21437/Eurospeech.1999-92

ABSTRACT

A general class of Computationally Efficient locally Recurrent Networks (CERN) is described for real-time adaptive signal processing. The structure of the CERN is based on linear-in-the-parameters single-hidden-layered feedforward neural networks such as the Radial Basis Function (RBF) network, the Volterra Neural Network (VNN) and the recently developed Functionally Expanded Neural Network (FENN), adapted to employ local output feedback. The corresponding learning algorithms are described and key structural and computational complexity comparisons are made between the CERN and conventional Recurrent Neural Networks. A speech signal is used, which shows that a Recurrent FENN based adaptive CERN predictor can significantly outperform the corresponding feedforward FENN and conventionally employed linear adaptive filtering models.

1. INTRODUCTION

In conventional multi-layered feedforward neural networks such as the Multi-Layered Perceptron (MLP), nonlinear transformations are performed on the data at both input and output layers thus requiring the use of complex non-linear learning algorithms (such as Back Propagation). This leads to problems such as slow convergence and multi-minimum error surfaces [1-3]. Optimisation techniques such as genetic algorithms, learning automata and simulated annealing, although capable of achieving the global minimum require extensive computations [2]. Recently linear-in-the-parameters (with respect to the output layer weights) neural networks using a single hidden layer, namely the Radial Basis Function (RBF) [2], the Volterra Neural Network (VNN) [3] and a newly developed Functionally Expanded Neural Network (FENN) [7][8] have been proposed to alleviate the need of non-linear learning algorithms.

The other main paradigm of neural networks, namely Recurrent Neural Networks (RNNs) [2][5][9] [11-13] have conventionally been based on multi-layered feedforward neural networks, with feedforward and feedback connections between some or all of the network nodes. Consequently, the main drawback of RNNs to-date has been their learning computational requirements. For example, the well known Real-Time-Recurrent Learning (RTRL) algorithm [9] which is widely used to update RNNs has a computational complexity which increases as $O(N^4)$, where N is the total number of network nodes [5].

In this paper, we present a class of RNNs based on feedforward linear-in-the-parameters neural networks, employing local output feedback. Specifically, by exploiting the links between recurrent linear-in-the parameters neural networks and conventional Infinite Impulse Response (IIR) filters, a recursive Gauss-Newton type and stochastic gradient algorithm are given that are capable of operating in real-time. It is shown that the computational requirements of these algorithms is significantly less, of $O(N)$ to $O(N^2)$, compared to $O(N^4)$ for the conventional RTRL.

The paper is organized as follows: section 2 presents the general structure of the proposed class of Computationally Efficient Recurrent Networks (CERN) together with an overview of the corresponding learning algorithms. Section 3 discusses the computational requirements of the proposed algorithm and compares them with those of the RTRL algorithm employed by conventional RNNs. Section 4 presents the results of applying the CERN to real-time modeling of speech signals. These results compare the performances of the recurrent FENN, the feedforward FENN and a conventional linear filter. Finally, section 5 presents the conclusions.

2. THE COMPUTATIONALLY EFFICIENT RECURRENT NETWORKS (CERN)

The general structure of the Multiple n Input Multiple m Output (MIMO) computationally efficient recurrent networks (CERN) is illustrated in Figure 1. As can be seen, it is basically a two-layer feedforward linear-in-the-parameters neural network employing local output feedback. The single hidden layer comprises a functional expander which performs a non-linear transformation that maps the input space onto a new non-linear hidden space of possibly increased dimension. The non-linear approximation ability of the CERN relies mainly on the type of functional expansion employed. The form of the functional expansion model is completely general, and depends on the choice of the feedforward ANN employed [7].

For the case of a FENN based CERN, the functional model uses a *hybrid* expansion that is, a combination of sigmoidal shaped, Gaussian shaped and polynomial subset activation functions [7].

The output layer of the CERN comprises a set of linear combiners corresponding to the number of desired outputs (available for network training). Each network output is the result of summation of the weighted values of the non-linearly expanded input terms and its own past values. Mathematically, the *j*-th output of the CERN (for a [1x*N*] functional expansion of the *n* inputs and feedback of M_j past values of the *j*-th output) is:

$$y_j(k)=\sum_{i=1}^{M_j} a_{ij}\, y_j(k-i)+\sum_{l=1}^{N} b_{lj}\, f_{lj}(k) \quad \text{for } j=1,2,...,m \text{ outputs} \tag{2.1}$$

where a_{ij} and b_{lj} are the feedback and feedforward weights respectively for the *j*-th output, $f_l(k)$, $l=1,...,N$ are the functionally expanded input terms. define, which transform the input space R^n of *n* inputs $[x_1(k)\ ...\ x_n(k)]$ onto a new non-linear hidden space of increased dimension R^N; and $y_j(k-i)$, $i=1,...,M_j$ are the M_j past values of the *j*-th output $y_j(k)$. The above CERN is said to be recurrent of order M_j for the *j*-th output $y_j(k)$. Note that each output layer node has feedback from the weighted past values of its own output only, not from the outputs of other nodes. In this context, the CERN employs local output feedback similar to Frasconi et al's architecture [4], as opposed to global output feedback employed in the fully recurrent structures such as the Real-Time Recurrent Network (RTRN) [9].

The weights for a general MIMO CERN are updated as follows:
Defining at time *k*: $\Theta_j(k)=[a_{1j}(k)\ ...\ a_{Mj}(k)\ b_{1j}(k)\ ...\ b_{Nj}(k)]^T$ and $X_j(k)=[y_j(k-1)\ ...\ y_j(k-M_j)\ \ f_1(k)\ ...\ f_N(k)]^T$ equation 2.1 above may be written as:

$$y_j(k)=\Theta_j^T(k)X_j(k) \quad \text{for } j=1,2,..., m \tag{2.2}$$

where T denotes the transpose operator. $\Theta_j(k)$ is now chosen to minimise the Mean Squared Error (MSE) cost function *J* defined as: $J=(1/2)\ E(e_j(k)^2)$, where $E(.)$ is the expectation operator, and $e_j(k)$ is the *j*-th output error defined by: $e_j(k)=d_j(k)-y_j(k)$, with $d_j(k)$ being the *j*-th desired output (which is available during the CERN on-line or off-line training mode). Using a gradient approach, a recursive estimator for the *j*-th output gradient with respect to all its feedforward and feedback weight coefficients can be identified as [7]:

$$\frac{\partial y_j(k)}{\partial \Theta_j(k)}= X_j(k)+\sum_{i=1}^{M_j} a_{ij}(k)\frac{\partial y_j(k-i)}{\partial \Theta_j(k-i)} \quad \text{for } j=1,...,m \tag{2.3}$$

where the recursive gradient estimator for each output is the following column vector:

$$\frac{\partial y_j(k)}{\partial \Theta_j(k)}=\left[\frac{\partial y_j(k)}{\partial a_{1j}(k)},...,\frac{\partial y_j(k)}{\partial a_{M_j j}(k)},\frac{\partial y_j(k)}{\partial b_{1j}(k)},...,\frac{\partial y_j(k)}{\partial b_{Nj}(k)}\right]^T$$

The gradient expressions derived above are exact and gradient search iterative techniques can now be used to approximate optimum $\Theta_j(k)$. Assuming the *instantaneous* gradient estimate, a Real-Time Recursive Update (RTRU) algorithm for a MIMO $(n,N;m,(M_1,M_2,...,M_m))$CERN (with *n* and *m* denoting the number of inputs and outputs respectively and $M_1,M_2,...,M_m$ representing the past M_j values of each of the $y_j(k-i)$ $i=1,..., M_j$, $j=1,...,m$ outputs fed back) can now be written as follows:

CERN-RTRU :

(1) Compute the CERN *m* outputs $y_j(k-i)$ $i=1,..., M_j$ for $j=1,...,m$, using Equation 2.2 above

(2) Compute each CERN output gradient estimate using Equation 2.3 above

(3a) Update CERN weights for each output using a recursive Gauss-Newton update algorithm [7]

or

(3b) A stochastic gradient LMS type algorithms as follows:
CERN-stochastic gradient:

$$\Theta_j(k+1)=\Theta_j(k)+\mu e_j(k+1)\frac{\partial y_j(k+1)}{\partial \Theta_j(k+1)} \quad \text{for } j=1,...,m$$

where μ is the convergence factor.

This algorithm requires only the order of $(M+N)$ operations but has a much slower convergence.

3. COMPUTATIONAL CONSIDERATIONS AND COMPARISON WITH THE CONVENTIONAL RTRL ALGORITHM

Conventional RNNs are trained by the RTRL algorithm [5] which is a temporal supervised learning algorithm based on an approximation to the method of steepest descent. The RTRL is *nonlocal* in the sense that each weight much have access to the complete weight matrix and the complete error vector. At any time *k* it requires consideration of a total of (M^3+MN^2) values of each dynamic output gradient computed with respect to all the feedback and feedforward weights [5]. On the contrary, the CERN learning algorithm RTRU is *local* and requires just (M^2+MN) operations for the estimation of each of its output gradient (assuming the same number of hidden layer nodes *N* in both the structures).

Note that the above reduction in the relative complexity of the CERN learning algorithms has been achieved by employing non-linear basis functions only at the single input hidden layer of the CERN, whereas in conventional multi-layered RNN structures non-linear basis functions are employed at both the hidden and output layers. Therefore, the CERN structure is similar to a simple IIR type filter with a functionally transformed input, whereas on the contrary, conventional RNNs are highly non-linear in the parameters Multi-Layered Perceptron (MLP) based structures with full interconnections (feedforward and feedback) between all nodes.

4. APPLICATION OF THE CERN TO REAL-TIME SPEECH MODELLING

As discussed by Connor et al [12] and Hussain [7], Recurrent ANNs (RANNs) will be more effective than Feedforward ANNs (FANNs) in modeling only certain classes of non-linear dynamical processes. Specifically, for the modeling of Non-linear Auto-Regressive Moving Average NARMA output-error type processes, RANN based non-linear predictors will be more effective than corresponding FANN based predictors. On the other hand, RANNs will not give any advantage over FANNs in the modeling of Non-linear Auto-Regressive NAR type time series processes. For the difficult problem of real-time adaptive non-linear prediction of non-stationary signals, which is synonymous with on-line modeling of the underlying physical mechanism responsible for the generation of the time series, the traditional method of supervised learning employed by the conventional FANNs and RANNs is unsuitable because of their off-line requirements [5]. What is needed is a neural network that can learn on-line that is, the network continuously learns to adapt to the statistical variations of the incoming time series whilst performing its filtering operation.

The CERN structure described above can be readily adapted to perform the above role by making the effective training period of the RTRU algorithm equal to infinity. Thus the training period is equated to the length of the incoming time series. In the following example, a real speech signal will be modeled using, a feedforward (2,20;1)FENN, a corresponding (2,20;1,1)RFENN based CERN, and a similar order linear FIR predictor

The overall MSEs computed for each predictor are compared in Table 1, which shows that the non-linear RFENN gives the best adaptive performance followed by the FENN and the FIR. In [5] Haykin suggests that the performance of a non-linear adaptive predictor may be enhanced by using a linear predictor in series with a RNN. A new hybrid RFENN-FIR based modified CERN predictor was thus devised, comprising a (2,20;1,1)RFENN subsection with its output feeding into a 16-tap linear FIR subsection. The weights of the FIR subsection were adapted using the Least Mean Squares (LMS) algorithm. In Table 2, the RFENN-FIR based adaptive predictor model is seen to significantly outperform the stand-alone RFENN based CERN structure.

	RFENN	FFENN	FIR	RFENN-FIR
MSE	0.0252	0.0264	0.030	0.0170

Table 1: Performance Comparison of non-linear 2nd order RFENN based CERN, FFENN and linear 20-th order FIR based adaptive Single-Step Predictors on a real speech signal.

5. CONCLUSIONS

A general class of CERN structures has been presented which are based on single-hidden layered, linear-in-the-parameters feedforward ANNs adapted to employ local output feedback. The associated RTRU learning algorithm has been derived and structural and learning computational requirements of the CERN shown to be significantly simpler compared to those of the conventional multi-layered Recurrent Neural Networks (RNNs). A RFENN based CERN has been successfully employed for performing adaptive (on-line) non-linear prediction of real-world noisy highly non-stationary time series, and shown to outperform the feedforward FENN and the conventional linear adaptive filtering models. In various other case studies detailed in [7], RFENN based predictor models have been employed for off-line modeling of NARMA output-error type time series processes, and shown to significantly outperform the FFENN and MLP based predictors, whilst providing comparable performance to the conventional more complex multi-layered fully RNN predictor models recently reported in [12].Multi-dimensional time series processes have also been successfully modeled using multiple output FENN based predictor models [7]. Details on the various possible structural and learning variations of the CERN and in particular, their relationship to a recently reported class of Locally Recurrent Globally Feedforward Networks (LRGFN) [13] can be found in [7]. Currently, the adaptive modeling performance of the RFENN is being compared with other CERN structures namely, the Recurrent RBF and Recurrent VNN, in addition to conventional RNN based predictor models, for a variety of real-world processes.

6. REFERENCES

[1] S.Chen, S.A.Billings, C.F.N.Cowan, P.M.Grant, *Parallel recursive prediction error algorithm for training layered neural networks,* International Journal of Control, Vol.51, No.6, pp.1215-1228, 1990.
[2] D.R.Hush, B.G.Horne, *Progress in Supervised Neural Networks,* IEEE Signal Proc. Mag., pp.9-39, January 1993.

[3] P.J.W.Rayner, *A new connectionist model based on a non-linear adaptive filter*, Proc. ICASSP, pp.1191-94, 1989.
[4] P.Frasconi, M.Gori, G.Soda, *Local feedback Multi-layered networks*, Neural Comp,Vol.4,pp.120-130,1992.
[5] S.Haykin, *Neural Networks - A Comprehensive Foundation*, MacMillan, 1994
[6] J.J.Shynk, *Adaptive IIR filtering*, IEEE ASSP Magazine, April1989.
[7] A.Hussain, *Novel Artificial Neural Network Architectures and Algorithms for Non-linear Dynamical System Modeling and Digital Communications Applications*, PhD Thesis, University of Strathclyde, Glasgow G1 1XW, U.K., 1996.
[8] A.Hussain, J.J.Soraghan, T.S.Durrani, *A new neural network for non-linear time series modeling,* International Journal of Computational Intelligence in Finance, Vol.5, No.1, pp.16-26, Finance and Technology Publishing, U.S.A, 1997.
[9] R.J.Williams, D.Zipser, *A learning algorithm for continually running fully recurrent neural network.* Neural Computation, Vol.1, no.2, pp.270-280, 1989.
[10] M.E.Salgado, G.C.Goodwin, R.H.Middleton, *Modified Least squares algorithm incorporating exponential resetting and forgetting*, Intern. Journal Control, Vol.47, pp.477-491, 1988.
[11] A.G.Parlos, K.T.Chong, A.F.Atiya, *Application of Recurrent Multi-Layered Perceptron in modeling complex process dynamics*, IEEE Trans. Neural Networks, Vol.5, no.2, pp.255-265, 1994.
[12] J.T.Connor, R.D.Martin, L.E.Atlas, *Recurrent Neural Networks and Robust Time Series Prediction,* IEEE Trans. on Neural Networks, Vol.5, pp.240-254, March1994.
[13] A.C.Tsoi, A.Back, *Locally Recurrent Globally Feedforward Networks: A critical Review of Architectures*, IEEE Trans. on Neural Networks, Vol.5, no.2, pp229-239, March1994

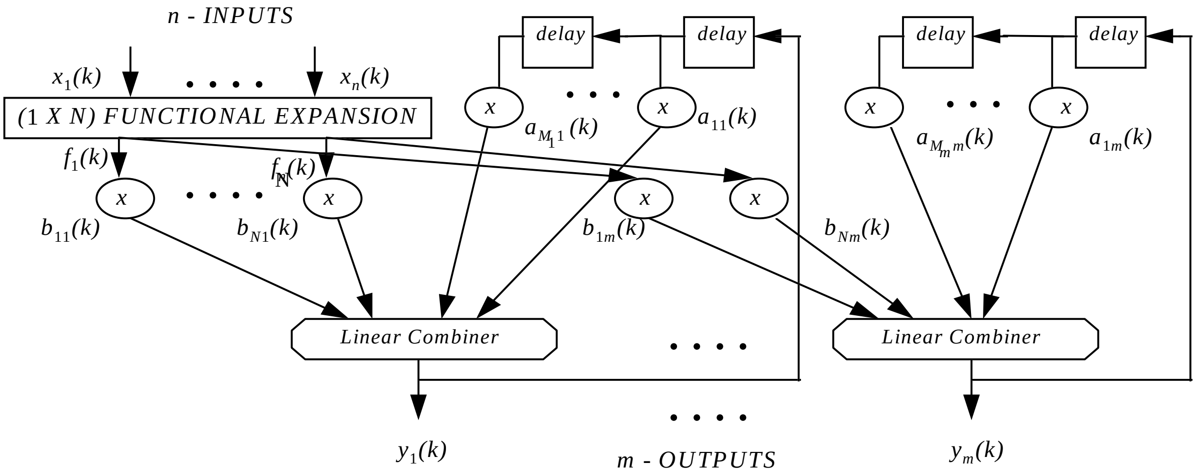

Figure 1: General Structure of the MIMO CERN.