APPLICATION OF RECURRENT NEURAL NETWORKS TO COMMUNICATION CHANNEL EQUALIZATION

M.J.Bradley and P.Mars

School of Engineering, University of Durham, South Road, Durham, DH1 3LT, UK.

ABSTRACT

This paper examines the mechanism by which Recurrent Neural Networks (RNNs) acheive equalization whilst operating on simple digital communication channels. The mode of operation is seen to be essentially similar to the conventional Decision Feedback Equalizer (DFE) and the RNN node nonlinearity is identified as a limiting factor. Two versions of an alternative RNN structure are formulated for channels with longer impulse responses based on soft-decision feedback. Simulations demonstrate the improved BER performance compared with the DFE.

1. INTRODUCTION

Recently a number of neural network structures have been applied to the problem of equalization of digital communication channels. The benefit of neural networks in this area lies in their pattern classification abilities which are used to distinguish between a finite set of transmitted symbols arriving at the receiver in a corrupted state. The corruption is primarily due to intersymbol interference and additive noise occuring in the communication channel. This contrasts with the more traditional method of treating equalization as an inverse filtering problem rather then a classification problem.

A number of network structures have been proposed: the Multilayer Perceptron (MLP) [1], the Radial Basis Function network (RBF) [2] and, more recently, the Recurrent Neural Network (RNN) [3]. Each network uses node nonlinearities to form nonlinear decision boundaries in order to differentiate between the transmitted symbols. Chen et al. [2] implement the optimal symbol-by-symbol equalizer, a maximum *a posteriori* (MAP) detector, as an RBF network with the node nonlinearities matching the channel noise probability density function. Kechriotis et al. [3] apply an RNN with a hyperbolic tangent nonlinearity to equalize linear and nonlinear channels.

This paper examines the mechanism by which simple RNN equalizers achieve equalization and identifies the node nonlinearity as a limiting factor. An improved RNN structure is formulated and compared with alternative equalizer structures.

The RNN structure is shown in Fig. 1 and consists of a delay line and a fully interconnected set of nodes. Each node forms a weighted sum of its inputs, the activation, and passes this signal through a hyperbolic tangent nonlinearity to produce the node output. The input to the network is the sampled received signal and both the delay lines and the feedback paths between nodes have delay elements equal to the transmitted symbol period. The output of one node is designated as the network output and this is passed to a decision device for classification. The outputs of the remainder of the nodes are not defined to perform any explicit function. The network is adapted using the Real-Time Recurrent Learning (RTRL) algorithm [4] which is a gradient-descent based algorithm, the error surface being formed from the instantaneous square error between the network output and the transmitted symbol. This error definition is suitable for PAM/QAM transmission schemes where the network output estimates the symbol amplitude. The training algorithm requires a training sequence to be transmitted during adaption of the network and attempts to minimise the squared error of the network output node.

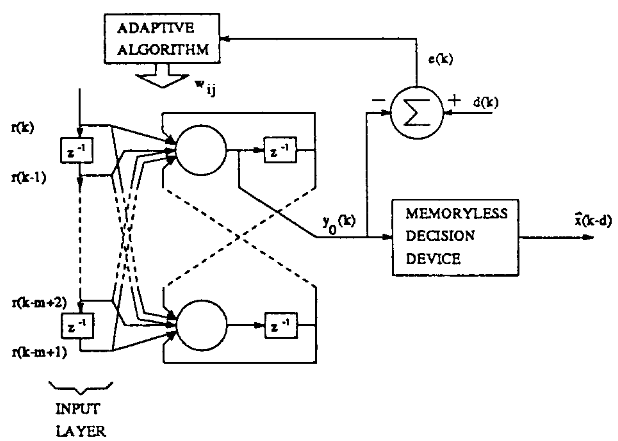

Figure 1: Structure of the Recurrent Neural Network equalizer.

2. SINGLE NODE DYNAMICS

Initial simulations of the RNN equalizer operating on a 2 path stationary channel conveying binary PAM data showed that the RTRL algorithm reduced the network to a single node structure using one received sample from the input delay line and a self-feedback term. This structure (Fig. 2) is effectively a nonlinear IIR filter which can exhibit a variety of dynamics dependent on the 2 node weights, w_a and w_b. After a step input, the RNN reaches an equilibrium state when

$$y(k) = y(k-1) = \tanh(w_a y(k-1) + w_b r(k)) \qquad (1)$$

0-7803-2431-5/95 $4.00 © 1995 IEEE

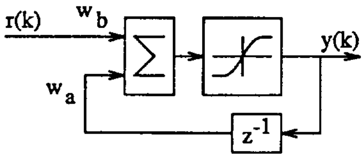

Figure 2: Structure of single node equalizer.

The step responses include a single stable equilibrium state when $|w_a| < 1$ and a combination of 1 or 2 stable equilibrium states plus an unstable equilibrium state when $|w_a| > 1$. Limit cycles/saturation occur if

$$\frac{d}{ds}\tanh(s) > \left|\frac{1}{w_a}\right| \quad (2)$$

at the activation, s, of the equilibrium states. As $|w_a|$ increases the node takes longer to converge to the equilibrium state/ limit cycle.

3. SINGLE NODE RNN EQUALIZER DESIGN

The RNN output is designed to be $\pm y_{eq}$, corresponding to the transmission of symbols ± 1. The RNN equalizer must be stable under a constant input (equivalent to the repeated transmission of either of the 2 symbols through a stationary channel with no additive noise), thus $\pm y_{eq}$ are the designed equilibrium states.

For a channel of the form $H(z) = h_0 + h_1 z^{-1}$ and zero equalizer decision delay, the constant input corresponding to the +1 symbol is $(r_0(k) = h_0 + h_1)$. Ideally the equalizer should also change from the $-y_{eq}$ state to the $+y_{eq}$ state in one iteration after receiving the sample $(r_1(k) = h_0 - h_1)$. Application of Eqn. 1 for these input conditions gives equations for the node weights:

$$w_a = -\frac{\tanh^{-1}(y_{eq}).h_1}{y_{eq}.h_0} \quad (3)$$

$$w_b = \frac{\tanh^{-1}(y_{eq})}{h_0} \quad (4)$$

The same equations are obtained by using the remaining 2 samples, $(r_2(k) = -h_0 + h_1)$ and $(r_3(k) = -h_0 - h_1)$, as input conditions for the transition to the $-y_{eq}$ state. Simulations show that the RTRL algorithm adapts the RNN weights to these values.

For certain nonminimum phase channels the required setting of the weight, w_a leads to limit cycles/saturation. Applying Eq. 3 to Eq. 2 with $(s = \pm\tanh^{-1}(y_{eq}))$ gives the value of the channel zero in which limit cycles occur:

$$\left|\frac{h_1}{h_0}\right| \geq \left|\frac{\exp(4s) - 1}{4s\exp(2s)}\right| \quad (5)$$

For longer channel impulse responses the RNN node must change from the $-y_{eq}$ state to the $+y_{eq}$ state for a larger set of received samples, $r_i(k)$. For the transition to occur in a single step, no solutions exist for the weights, w_a and w_b that are independent of the received samples. The single step transition must either be relaxed or the transition to a set of points in the vicinity of the desired equilibrium state must be permitted.

Additive channel noise causes the node output to vary from the desired equilibrium states. If the noise is such that the output changes sign then a classification error will occur (the decision device being a zero-threshold slicer for binary PAM symbols). An approximation of the probability of error may be obtained by calculating the magnitudes of the noise samples, n_i, that cause a classification error for each of the received samples, $r_i(k)$, $i = 0, \ldots, 3$. Substituting $(r_i(k) + n_i)$ for $r(k)$ in Eqn. 1 and using the condition that $y(k)$ is of the opposite sign to the desired output, classification errors occur when $(n_i > h_0)$. Thus, for a zero mean Gaussian noise pdf, the probability of error is given by

$$P_e = \frac{1}{2}\operatorname{erfc}\left(\frac{h_0}{\sqrt{2}\sigma}\right) \quad (6)$$

This approximation assumes that the output is initially at one of the designed equilibrium states and, in general, will not be valid due to the prescence of noise on previously received samples. Noise on one sample adversely affects the equalizer performance on subsequent samples and Eqn. 6 becomes a lower bound on the probability of error.

The problems of the noise feedback mechanism and the limitations on nonminimum phase channels and channels with longer impulse responses are removed if the tanh(·) nonlinearity is replaced by a slicer. There is more freedom in weight specification as there is no unique value of activation that gives the desired output states. Only channel noise that results in classification errors will result in noise being fed back to affect subsequent outputs. The resulting structure is precisely that of a DFE with a single feedback tap. From Eqns. 3 and 4, the feedback weight is seen to cancel the intersymbol interference created by the channel as per the DFE [5].

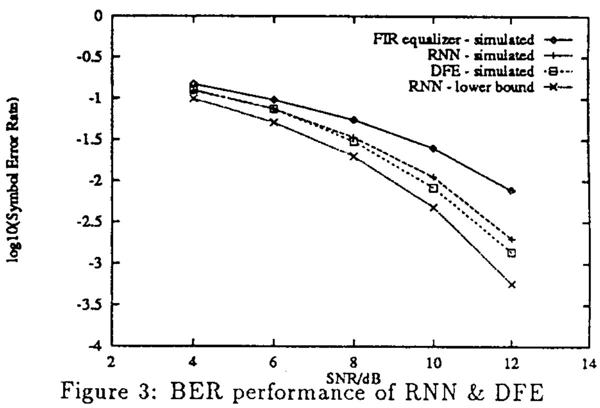

Figure 3: BER performance of RNN & DFE

Simulations comparing the bit error rate (BER) for the channel $H(z) = 0.8192 + 0.5734z^{-1}$ show the performance improvement over the single node RNN obtained using a DFE with 1 input tap, 1 feedback tap and zero decision delay (Fig. 3). The figure also shows the RNN lower probability of error bound (Eqn. 6).

4. DESIGN OF MULTINODE RNNS

Similarities between the DFE and RNN were observed in a number of simulations of multinode RNNs operating on longer channels. In some cases the network trained to a structure such that nodes, with the exception of the output node, were being used merely as symbol-period delays. Furthermore the weights on the connections from these nodes to the output node were such that they cancelled the interfering symbol terms of the filter obtained by the convolution of the channel filter and the FIR filter formed by the output node from the received sample delay line. Again the RNN is operating in the same manner as the DFE by cancelling the post-cursors of the effective channel impulse response [5].

The multinode RNN may be reconfigured to use the slicer nonlinearity and with node i estimating the symbol, $x(k-d-i)$, where d is the decision delay of the equalizer and $(i = 0, 1, \cdots)$. This structure (Fig. 4) is similar to a bank of DFEs operating in parallel, each with a differing decision delay and using the outputs to provide old symbol estimates rather than a feedback delay line. This allows symbol estimates to be updated as and when more of the transmitted symbol energy arrives at the receiver (i. e. soft decision feedback) and to potentially reduce the problem of error propagation. This method is only suitable for equalizers operating with a decision delay less than the time span of the channel impulse response.

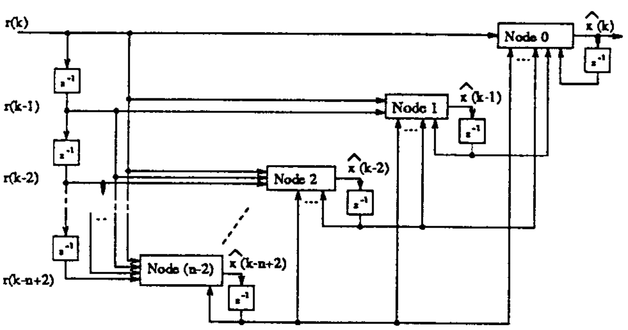

Figure 4: Structure of the modified RNN equalizer.

Connections to a node from the received sample delay line only exist for samples containing energy from the transmitted symbol that the node is estimating. Feedback connections are restricted such that only interfering terms due to symbols transmitted prior to the symbol being estimated by the node are cancelled (the channel post-cursors). The number of nodes in the network is $(n_c - 1)$ for a channel with an impulse response of n_c samples. This provides only node 0 with complete cancellation of the channel post-cursors and so an additional delay line fed by symbol estimates from node $(n_c - 2)$ is used to provide symbol estimates not generated explicitly by nodes. This results in node $(n_c - 2)$ having exactly the same structure as the DFE.

Adaption of the new network is achieved using the LMS algorithm in conjunction with a training signal. The algorithm requires that nodes have correct symbol feedback during training. This makes the nodes independent of each other and allows a seperate error signal and adaption adaption algorithm to be formed for each node. As in the RTRL algorithm, the error metric is the instantaneous square error between the transmitted and estimated symbol. The node activation is taken as the symbol estimate rather than the node output due to the undifferentiable node nonlinearity.

An alternative RNN (structure 2) may be formed using additional connections from nodes with smaller decision delay to cancel channel pre-cursors and creates a structure that is more fully recurrent. This structure has an improved performance given correct symbol feedback as a greater proportion of the intersymbol interference is cancelled. The adaption algorithm is modified to account for the additional feedback taps.

5. PERFORMANCE COMPARISON

The performance of the 2 alternative RNN structures were compared against the DFE by simulating the bit error rates as function of signal-to-noise ratio whilst operating on the stationary nonminimum phase channel, $H(z) = -0.2052 - 0.5131z^{-1} + 0.7183z^{-2} + 0.3695z^{-3} + 0.2052z^{-4}$ corrupted by additive white Gaussian noise.

All simulations used a 2 level PAM scheme for the transmitted symbols and the equalizers were trained for 2000 symbols using LMS type adaption algorithms prior to BER measurement. A step size of $\mu = 0.05$ was used in all adaption algorithms. After each adaption process was completed, 10^4 symbols were transmitted and the number of errors accumulated. This process was repeated until 10^5 or 10^6 nontraining symbols had been transmitted and an ensemble averaged BER calculated. This policy was adopted so that the variations in the weight settings inherent in the LMS adaption algorithm are averaged.

Both RNN structures had 4 nodes and inputs of up to 4 received channel samples with a decision delay, $d = 0$. A feedback delay line of length 3 fed by node 3 was used to generate the symbol estimates $\{\hat{x}(k-5), \ldots, \hat{x}(k-7)\}$. The 4 comparable DFE structures had $m = (d+1)$ received sample inputs where the decision delay, $d = 0, \ldots, 3$ and each used $n = 4$ past decisions to cancel intersymbol interference. The different DFE structures were used to compare the performance of the output of each of the 4 nodes in the RNN structures.

Figure 5 shows the performance of a DFE operating with $d = 0$ against the 2 RNN structures using node 0 as the equalizer output. The differences between the 2 RNN structures is negligible. The lower curve shows the performance of all 3 equalizer structures operating with correct symbol feedback. This curve is the same for both RNN and DFE equalizers as the input configuration of node 0 of the RNNs and of the DFE is identical. The improved performance of the modified RNN equalizers over the DFE is clear.

Figure 6 compares the pertormance of a DFE with $d = 1$ against the RNN structures using node 1 as their output. RNN structure 1 operates just below the optimum at high levels of SNR. The correct symbol feedback curve for all 3 equalizers is again the same due to the identical input configuration under these conditions.

Figure 7 indicates that for $d = 2$ the DFE and RNN structure 1 have similar performance. RNN structure 2 has a slightly larger BER at higher levels of SNR. This plot shows 2 correct symbol feedback curves — the first for RNN structure 1 and the DFE and the second for RNN structure 2. Node 2 of RNN structure 2 uses the output of node 0 as a symbol feedback input and so cancels a greater portion of the intersymbol interference occuring in its received sample inputs. Under correct symbol feedback conditions this makes RNN structure 2 more able to classify the recieved samples correctly. It is noted that the performance gap between ideal and actual is much greater for RNN structure 2 than RNN structure 1 and the DFE.

Finally, Figure 8 shows the performance of the structures with $d = 3$. The RNN structure 1 and the DFE have the same structure and so their performance is identical. RNN structure 2 has a consistently poorer BER compared to RNN structure 1 and the DFE. Once more the deviation between ideal and actual is greater for RNN structure 2.

6. CONCLUSIONS

The RNN equalizer has been observed to reduce intersymbol interference by a method similar to the DFE. For 2 path channels the network reduces to a single node and the use of hyperbolic tangent nonlinearity is seen to increase the network's susceptibility to noise. Theoretical settings of the network weights are formulated and are observed in networks adapted by the RTRL algorithm.

Two alternative RNN structures have been developed, both using slicer nonlinearities and with each node output explicitly defined. For a comparable decision delay, a performance improvement of RNN structure 1 over the DFE is indicated. RNN structure 2 has a generally inferior performance due to incorrect symbol classification by low decision delay nodes increasing rather than decreasing intersymbol interference present on other nodes. It is concluded that RNN structure 1 outperforms the DFE only when minimal decision delay can be tolerated.

7. REFERENCES

[1] G.J.Gibson, S.Siu, and C.F.N.Cowan. Multilayer perceptron structures applied to adaptive equalisers for data communications. In *ICASSP-89*, 1183–1186, 1989.

[2] S.Chen, B.Mulgrew, and S.McLaughlin. Adaptive Bayesian equalizer with decision feedback. *IEEE Trans. Sig. Proc.*, 41(9):2918–2927, 1993.

[3] G.Kechriotis, E.Zervas, and E.S.Manolakos. Using recurrent neural networks for adaptive communication channel equalization. *IEEE Trans. Neural Networks*, 5(2):267–278, 1994.

[4] R.J.Williams and D.Zisper. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1:270–280, 1989.

[5] J.Salz. Optimum mean-square decision feedback equalization. *Bell System Technical Journal*, pages 1341–1373, October 1973.

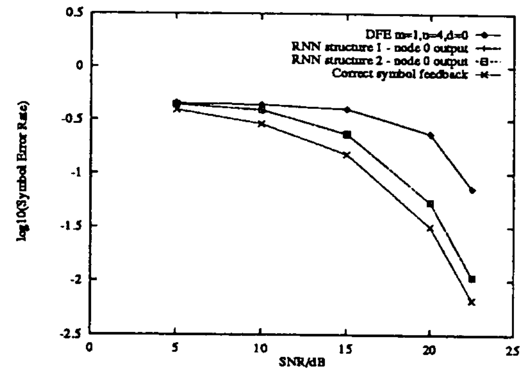

Figure 5: BER performance of RNN structures 1 & 2 and DFE with $d = 0$.

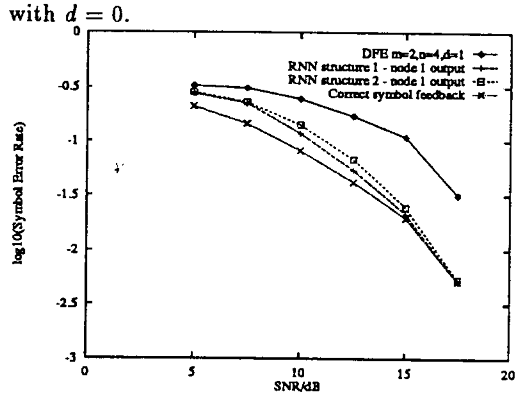

Figure 6: BER performance of RNN structures 1 & 2 and DFE with $d = 1$.

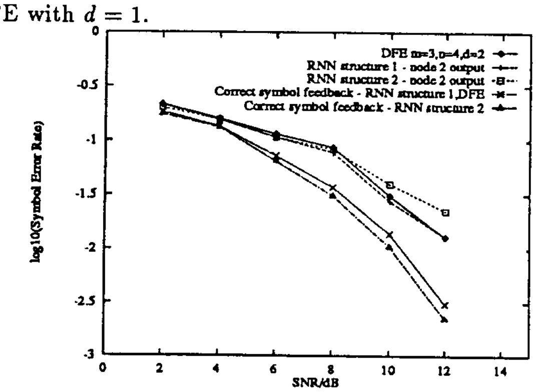

Figure 7: BER performance of RNN structures 1 & 2 and DFE with $d = 2$.

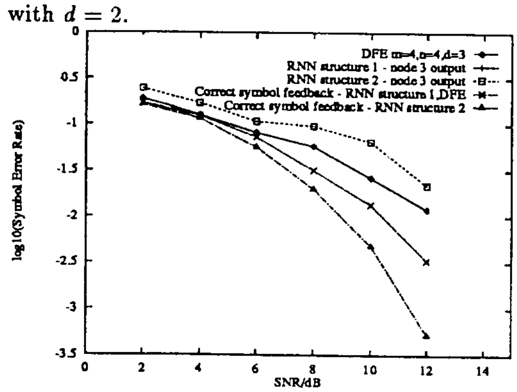

Figure 8: BER performance of RNN structures 1 & 2 and DFE with $d = 3$.