

Craven, Michael P. and Curtis, K. Mervyn and Hayes-Gill, Barrie R. (1996) Multiple channel crosstalk removal using limited connectivity neural networks. In: 3rd IEEE International Conference on Electronics, Circuits, and Systems (ICECS 96), 13-16 October 1996, Rhodes, Greece.

Access from the University of Nottingham repository:

<http://eprints.nottingham.ac.uk/1903/1/icecs96crosstalk.pdf>

Copyright and reuse:

The Nottingham ePrints service makes this work by researchers of the University of Nottingham available open access under the following conditions.

This article is made available under the University of Nottingham End User licence and may be reused according to the conditions of the licence. For more details see:

http://eprints.nottingham.ac.uk/end_user_agreement.pdf

A note on versions:

The version presented here may differ from the published version or from the version of record. If you wish to cite this item you are advised to consult the publisher's version. Please see the repository url above for details on accessing the published version and note that access may require a subscription.

For more information, please contact eprints@nottingham.ac.uk

MULTIPLE CHANNEL CROSSTALK REMOVAL USING LIMITED CONNECTIVITY NEURAL NETWORKS

M.P. Craven, K.M. Curtis and B.R. Hayes-Gill

University of Nottingham, Department of Electrical and Electronic Engineering, University Park,
Nottingham, NG7 2RD, UK. Tel: +44 115 9515536 Fax: +44 115 9515616 E-mail: mpc@eee.nott.ac.uk

ABSTRACT

Limited connectivity neural network architectures are investigated for the removal of crosstalk in systems using mutually overlapping sub-channels for the communication of multiple signals, either analogue or digital. The crosstalk error is modelled such that a fixed proportion of the signals in adjacent channels is added to the main signal. Different types of neural networks, trained using gradient descent algorithms, are tested as to their suitability for reducing the errors caused by a combination of crosstalk and additional gaussian noise. In particular we propose a single layer limited connectivity neural network since it promises to be the most easily implemented in hardware. A variable gain neuron structure is described which can be used for both analogue and digital data.

1. INTRODUCTION

This paper is concerned with the use of mutually overlapping sub-channels for communication of multiple signals in electronic systems. Such a method was originally intended for the task of reducing the pin-out in VLSI inter-chip connections using Frequency Division Multiplexing (FDM), since the signals generated by a set of outputs may be combined on-chip and transmitted off-chip via a single wire. However, the method is equally applicable to the general case of crosstalk in multiple channel communications, with or without FDM. Previous work carried out at the University of Nottingham has demonstrated that multiple sub-channel analogue FDM is feasible only if deliberate overlapping of the sub-channels is allowed^[1]. This overlapping inevitably introduces adjacent channel (or cross-channel) interference which reduces the SNR in the receiver unless steps are taken to remove it.

A multiple channel communication model is described by Van Etten^[2] which uses an $M \times M$ interference matrix so that received signals are linear combinations of the wanted and interfering signals. In this paper the interference is considered to be resulting entirely from nearest adjacent sub-channels and the amount of interference is a fixed fraction ϵ of each of the adjacent signals, which is called the fractional crosstalk (or fractional overlap) parameter. This situation would arise in an FDM channel when receiver filter responses are overlapped by equal amounts, or in some kinds of multicore cable. It is necessary to examine the increase in error in the received signals caused by increasing this crosstalk, and thus compare methods for reducing the errors.

Consider a channel with M transmitters with signals denoted by the vector \mathbf{x} at a particular time. Thus the M

received signals denoted by the vector \mathbf{y} are given by $\mathbf{y} = R \mathbf{x}$ where R is the interference matrix of form,

$$R = \begin{pmatrix} 1 & \epsilon & 0 & 0 & 0 \\ \epsilon & 1 & \epsilon & 0 & 0 \\ 0 & \epsilon & 1 & \epsilon & 0 \\ 0 & 0 & \epsilon & 1 & \epsilon \\ 0 & 0 & 0 & \epsilon & 1 \end{pmatrix} \quad (1)$$

shown here for $M=5$.

2. NEURAL NETWORK ERROR REDUCTION

The aim is to find the inverse of R in order to regain (or equalise) the original signals. This can be written in the form of a single layer feedforward neural network, where the elements of the inverse matrix are approximated as closely as possible by the weights of the network i.e. $\mathbf{z} = W \mathbf{y}$ where $W \approx R^{-1}$, so $\mathbf{z} \approx \mathbf{x}$. Direct calculation of the inverse is possible, and a result is obtained analytically except for certain values of ϵ which make the matrix singular. The weights w_{ji} are also readily calculated iteratively by the μ -LMS algorithm^[3] as follows,

$$\Delta w_{ji} = 2\mu (x_i - z_i) y_j \quad (2)$$

However, such a network requires M^2 weights so a hardware architecture based on this scheme would scale rather badly with M . It has been demonstrated^[4] that the area of a fully connected neural network architecture scales with order M^3 . The use of *limited connectivity* neural networks are thus proposed here where many of the weight matrix elements are zero.

2.1. Limited connectivity architecture

A limited connectivity architecture simplifies the network and allows the number of weights to be scaled linearly with M , which is important both in reducing the overall size of network and in providing a extendable modular structure. In the case where connectivity is limited to three receivers, the number of weights required is $\approx 3M$ and the optimum weight matrix has the form,

$$W = \begin{pmatrix} w_0 & w_{+1} & 0 & 0 & 0 \\ w_{-1} & w_0 & w_{+1} & 0 & 0 \\ 0 & w_{-1} & w_0 & w_{+1} & 0 \\ 0 & 0 & w_{-1} & w_0 & w_{+1} \\ 0 & 0 & 0 & w_{-1} & w_0 \end{pmatrix} \quad (3)$$

which is a limited connectivity neural network with 3 inputs per neuron. Clearly, W will only be an

approximation to R^{-1} . For clarity, the signal to be corrected is given index 0 and the other transmitted signals are indexed ± 1 and ± 2 . The input signals to each neuron y_j comprise the received signal from the original receiving node and its nearest neighbours, which are then multiplied by the weights w_i and summed to give output z_0 . Thus, a maximum of 5 transmitted signals x_i have an effect on any one received signal.

2.2. Analogue data

Error correction can be carried out independently for each neuron by finding its optimal weights. In the case of analogue signals, the output must follow the amplitude of the corresponding transmitted signal node as closely as possible, in spite of the crosstalk errors introduced, for an arbitrary combination of signals from the 5 transmitting nodes. In this implementation the five transmitted values were random numbers in a uniform distribution of range $[0, V]$ (representing an analogue voltage range of 0 to V volts). Before the correction network the error is,

$$(y_0 - x_0) = \varepsilon x_{-1} + \varepsilon x_{+1} \quad (4)$$

and at the output of the neuron, the error expression is,

$$(z_0 - x_0) = (w_0 + \varepsilon w_{-1} + \varepsilon w_{+1} - 1)x_0 + (w_{-1} + \varepsilon w_0)x_{-1} + (w_{+1} + \varepsilon w_0)x_{+1} + \varepsilon w_{-1}x_{-2} + \varepsilon w_{+1}x_{+2} \quad (5)$$

The expectation $E[Z_0 - X_0]$ and variance $\text{Var}[Z_0 - X_0]$ can then be derived and minimised (by partial differentiation with respect to w_0 and w_1 , setting $w_{+1} = w_{-1} = w_1$ since the mean input vector is symmetrical) to yield the optimal values of w_0 and w_1 which are found to be,

$$w_0 = \frac{(3 - 2\varepsilon - \varepsilon^2)}{(1 + 2\varepsilon)(3 - 8\varepsilon + 7\varepsilon^2)} \quad w_1 = \frac{\varepsilon(4\varepsilon - 3)}{(1 + 2\varepsilon)(3 - 8\varepsilon + 7\varepsilon^2)} \quad (6)$$

The weights may also be found iteratively as before by approximate steepest descent using the μ -LMS rule^[3], computing the standard error at the output of the central neuron in the row as,

$$SE = \sqrt{\frac{1}{N} \sum_{s=1}^N (x_{0s} - z_{0s})^2} \quad (7)$$

where N is the number of samples used.

Software was designed to simulate the crosstalk and correction technique using the μ -LMS rule, coded in C. A neuron was trained on 500 sets of random signal samples for each value of fractional crosstalk from 0.0 to 0.5, using a learning rate μ of 0.005. Then, the generalisation performance of the network was measured for a further 500 sets of samples. Curves of standard error for various degrees of fractional crosstalk are shown in Fig 1, for an input range $[0, 5]$. The top trace is the error before the network i.e. if a neural network is not used. This error increases linearly as the amount of crosstalk is increased, as predicted.

The middle pair of curves show the error achieved for the 3-input neuron, which is significantly less than that before correction. The ability of the network to reduce the error is best for smaller fractional crosstalk. With $\varepsilon = 0.1$ the error reduction is 26.3dB. Comparison with theory shows that the μ -LMS rule is able to achieve nearly the optimum set of weights as obtained from equation 6.

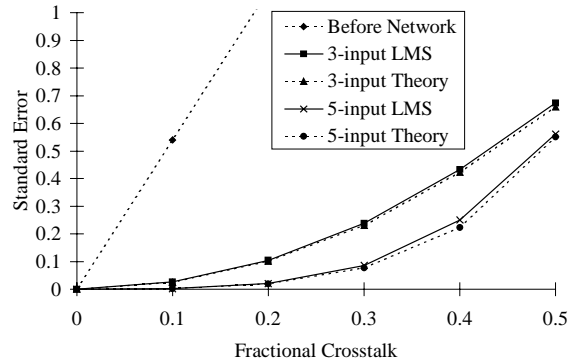


Fig 1. Crosstalk errors after training 3-input and 5-input neurons, compared with theory

The results for a further investigation using a 5-input neuron (lower curves in Fig 1) were found to be even better than those in the 3-input case with an error reduction of 48.6dB for $\varepsilon = 0.1$. The improvement is due to removal of the 'error-in-the-error' by the extra weighted inputs. A numerical computation of the minimum showed that the μ -LMS algorithm was again able to achieve nearly the optimum set of weights. Simulations with 7 and 9 input neurons were also carried out and the error was reduced even further. It is noted, however, that the use of a larger number of synapses is likely to increase the amount of random noise in a hardware implementation, and complicates the system by increasing the number of synapses needed. Therefore the 3-input neuron is to be preferred in practice.

2.3. Analogue data with additional random noise

We now consider the case where random noise is introduced into each of the transmitters such that we replace x_i by $(x_i + n_i)$, where n_i is a sample of gaussian noise added to transmitted sample x_i . We can now investigate the combined effects of crosstalk and noise. It is useful to define a Signal-to-Noise Power Ratio (SNR) at the output which includes the crosstalk as a type of noise. Thus,

$$SNR_{(dB)out} = 20 \log (V_{(rms)signal} / V_{(rms)noise}) \quad (8)$$

Although the noise in each transmitter is independent, the output from the 3-input neural network is a weighted sum from 3 different receivers, so some of the random noise will be correlated at the output. However, the amount of correlation will depend on the values of the weights. For small amounts of crosstalk requiring small weights w_{-1} and w_{+1} and a value of $w_0 = 1$, the amount of correlated noise will be small and the output SNR will be similar to that of the noiseless case. With

an input signal of peak amplitude 5v and a root-mean-square (RMS) value of $V_{\text{inputnoise}} = 0.01\text{v}$, the input SNR is 51.0dB, corresponding to 0.94 bits of error in a signal coded using 8 bits, which is a reasonable resolution for an analogue signal. Results were generated in the same way as before, this time using 5000 samples of signal plus noise. As seen in Table 1 the crosstalk dominates such that the output SNR is virtually unchanged for all but the smallest fractional crosstalk values, where random noise dominates.

Fractional Crosstalk	$\text{SNR}_{(\text{dB})\text{out}} V_{\text{inputnoise}}=0\text{v}$	$\text{SNR}_{(\text{dB})\text{out}} V_{\text{inputnoise}}=0.01\text{v}$
0.0	-	51.0
0.1	43.1	42.5
0.2	31.0	31.0
0.3	23.8	23.8
0.4	18.5	18.5
0.5	14.6	14.6

Table 1. Signal-to-noise ratios for crosstalk only, and crosstalk plus gaussian noise (RMS 0.01v)

$V_{\text{inputnoise}} / \text{Volts}$	0.1	0.2	0.3	0.4	0.5
$\text{SNR}_{(\text{dB})\text{in}} / \text{dB}$	31.0	24.9	21.4	18.9	17.0
Fractional Crosstalk	$\text{SNR}_{(\text{dB})\text{out}} / \text{dB}$ (for the above input noise values)				
0.0	31.0	25.1	21.6	19.2	17.4
0.1	30.5	24.7	21.3	19.0	17.2
0.2	27.4	23.1	20.1	18.0	16.4
0.3	22.6	20.3	18.1	16.4	15.1
0.4	18.1	16.9	15.6	14.5	13.6
0.5	14.5	14.0	13.3	12.7	12.1

Table 2. Signal-to-noise ratios for larger amounts of input noise (RMS 0.1-0.5v)

The next set of results, shown in Table 2, illustrates the effect of larger amounts of random noise (larger than would normally be encountered in an analogue system). It is seen that the larger amounts of random noise account for a significant lowering of SNR at the output for ϵ between 0.0 and 0.2, and a moderate reduction in SNR for $\epsilon=0.3$, whereas the loss due to random noise is at most 4.5dB for $\epsilon=0.4$ and higher. It was also noted that the absolute values of the weights were decreased as $V_{\text{inputnoise}}$ was increased, for any fixed value of fractional crosstalk.

2.4. Binary data

The previous results were obtained for transmission of analogue data with a uniform distribution. If data is digital, the multiplexed communication may use On/Off Keying (OOK) or Binary Phase Shift Keying (BPSK) modulation schemes. Equalisation in a BPSK system has been investigated by Wilson and Cioffi^[5] so OOK is considered here where the transmitted symbols are in the set $\{0, V\}$. The neural network architectures considered here are required to produce a digital output. Therefore, a neuron can be thought of as a series combination of a linear or non-linear neuron transfer function and a quantiser (hard-limiter). However, it is more useful to examine the error reduction before the quantiser, since hard-limiting the output masks the relative noise tolerances. If the interference matrix for adjacent channel interference is the same as before, the inputs to the neurons are in the set $\{0, \epsilon, 2\epsilon, 1, 1+\epsilon, 1+2\epsilon\}$. Thus, the 32 possible input patterns and corresponding outputs give rise to 17 unique non-trivial error expressions $E(\epsilon, w_1, w_0)$ to be minimised. Solving for zero error is impossible since it is not possible to satisfy all the error equations $|E(\epsilon, w_1, w_0)|=0$ simultaneously. However, it is possible to satisfy inequalities $|E(\epsilon, w_1, w_0)| < m$ for certain values of ϵ , where m is an acceptable margin of error greater than zero. Training was carried out as before for different values of ϵ , using the μ -LMS algorithm with $\mu=0.005$ and 500 training samples. Errors were measured at the network input, at the network output and after hard-limiting the output to 0 (if $z_0 \leq V/2$) or V (if $z_0 > V/2$), which is analogous to using $m=0.5$ above. Initial results showed that error reduction was possible up to around $\epsilon=0.3$, but the reduction was nothing like as good as for continuous inputs, and for $\epsilon=0.3$ the error was found to be worse than if the neuron is not used. Hard-limiting the outputs, however, had the result of reducing the error to zero for values of ϵ up to 0.3 whereas hard-limiting before the neuron was only effective up to $\epsilon=0.2$. Closer examination of the standard error during training showed that the network had not converged satisfactorily after 500 epochs, so it was necessary to increase the number of training epochs. These initial results also suggested that other neuron structures might be better at reducing the error.

The set of curves in Fig. 2 shows the standard errors obtained for four different networks with increasing amounts of fractional crosstalk, using 30000 training samples and a learning rate of 0.005; a linear limited connectivity network with 3 inputs per neuron, as above, then a network of with an extra bias input per neuron, denoted by vector \mathbf{b} , such that $\mathbf{z} = W \mathbf{y} + \mathbf{b}$, a non-linear network with $\mathbf{z} = F(W \mathbf{y} + \mathbf{b})$, where $F(x)=1/(1+e^{-x})$, and a 3-3-1 multilayer perceptron (MLP) network consisting of two non-linear layers. Also shown is the error obtained if no network is used. The two linear networks were trained using the μ -LMS algorithm, the single layer non-linear network was trained with the Delta Rule and the MLP was trained with the backpropagation (BP) algorithm (also called Generalised Delta Rule) respectively. The Delta Rule simply involves multiplying the previous weight change equation (2) by $F'(z_i) = z_i(1-z_i)$. The well known BP

algorithm for MLP neural networks is described in the literature^[6].

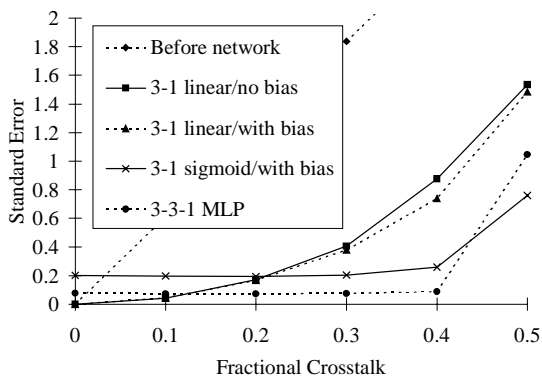


Fig 2. Separation of binary signals using various supervised neural networks

All networks tested were found to reduce the error compared with the situation where no network was used (straight line in Fig 2). Increasing the number of training epochs (30000) improved convergence in the 3-input limited connectivity case compared to the previous simulation with only 500 epochs, which shows that convergence is achieved more slowly than with analogue data with the same learning rate. Addition of a bias gave a small improvement in the case of the linear network, although this requires $\approx 4M$ weights compared to $\approx 3M$ without the bias. For fractional crosstalk up to 0.2, both linear networks have a slightly lower error than the non-linear ones. However, a combination of bias and sigmoidal function reduces the error considerably using a limited connectivity network with $\approx 4M$ weights. The best overall error reduction was obtained for the 3-3-1 multilayer network, where errors were small for all values of crosstalk. However, the number of weights required for the multilayer network is $\approx 16M$. In practice, for a hardware system, neither a multilayer nor a fully connected network are likely to be feasible, and after hard-limiting, the limited connectivity network has as good a performance as either of these for fractional crosstalk up to 0.4.

2.5. Binary data with additional random noise

A system with binary data is inherently more tolerant to noise since it is acting as a classifier. We considered On/Off keying of sinusoidal signals of amplitude $V=5v$, with $V_{inputnoise}$ ranging from 0.0v to 0.5v. The neural network used was the best previously considered single layer limited connectivity network, using a three input neuron with bias weight (i.e. 4 weights per neuron) and sigmoidal output. The output was then thresholded to binary values. At the output, the Bit Error Rate (BER) was computed based on 30000 samples, as defined by $BER = \text{Number of errors}/30000$, using a decision threshold of $z_0 = 2.5v$.

BER was found to be zero for $\epsilon \leq 0.2$ for all noise values considered. With $\epsilon=0.3$, BER was zero for $V_{inputnoise} \leq 0.3v$, 0.0001 at 0.4v and 0.0006 at 0.5v. With $\epsilon=0.4$, BER was zero for $V_{inputnoise} \leq 0.2v$, 0.002 at 0.3v, 0.004 at 0.4v and 0.01 at 0.5v. The cases with $\epsilon \geq 0.5$

were not resolvable even in the noiseless case with a $BER > 0.03$ for all noise values, since the neural network cannot define suitable decision boundaries. As crosstalk is increased, the position of the decision boundaries becomes more critical so that a decision is more greatly affected by the random noise. Of course, the situation may be improved by suitable coding and error correction techniques.

3. CONCLUSIONS

We have used limited connectivity architectures for crosstalk removal, such that the number of weights scales linearly with the number of signals. Each sub-channel has its own neuron which can be trained independently of the others. Thus a modular hardware implementation of the network could include separate learning hardware and weight storage for each neuron. Whilst a simple three weight neuron with a linear transfer function was adequate for error reduction with analogue signals, it is seen that an extra weight was required for biasing the transfer function in the binary case, and a sigmoidal output was also necessary to train the network with the Delta Rule. For binary data it is desirable to use hard-limiting outputs after the weights have been trained. The need to accommodate both types of signal suggests the use of a 3-input neuron with bias and a variable transfer function, which can be either linear, sigmoidal or hard-limiting. All these types of transfer function can be implemented using a variable gain sigmoid function $F(x) = 1/(1+e^{-kx})$, which has an almost linear response for small k and an almost hard-limiting response for large k .

ACKNOWLEDGEMENT

The funding for this work was provided by the UK Engineering and Physical Sciences Research Council, grant GR/J16701.

REFERENCES

- [1] CRAVEN M.P., CURTIS K.M. and HAYES-GILL B.R. : "Consideration of multiplexing in neural network hardware", IEE Proceedings Part-G, Circuits Devices and Systems, pp237-240, June 1994.
- [2] VAN ETEN W. : "An optimum linear receiver for multiple channel digital transmission systems", IEEE Trans. Communications, pp828-834, August 1975.
- [3] WIDROW B. and LEHR M.A. : "30 years of adaptive neural networks: Perceptron, Madaline, and Backpropagation", Proc. IEEE, 78,(9), pp1415-1442, September 1990.
- [4] BAILEY J. and HAMMERSTROM D. : 'Why VLSI implementations of associative VLCN require connection multiplexing', Proc. IEEE Int. Conf. Neural Networks, San Diego CA, Vol 2, pp173-180, 1988.
- [5] WILSON S.K. and CIOFFI J.M. : "Multi-dimensional equalisation for adjacent channel interference", IEEE International Conference on Communications (ICC '91), Vol 3, pp1398-1402, 1991.
- [6] RUMELHART D.E. and McCLELLAND J.L. (Eds.) : "Parallel Distributed Processing : Explorations in the microstructure of cognition", Vol. I & II (MIT Press, Cambridge MA, 1986).

