On-line learning algorithms for neural networks with IIR synapses

Paolo Campolucci, Francesco Piazza, Aurelio Uncini

Dipartimento di Elettronica e Automatica, Università di Ancona, via delle Brecce Bianche, 60131 Ancona, Italy,
Tel/Fax: +39 [71] 2204453 / 2204464, E-mail: paolo@eealab.unian.it

ABSTRACT

This paper is focused on the learning algorithms for dynamic multilayer perceptron neural networks where each neuron synapsis is modelled by an infinite impulse response (IIR) filter (IIR MLP). In particular, the Backpropagation Through Time (BPTT) algorithm and its less demanding approximated on-line versions are considered. In fact it is known that the BPTT algorithm is not causal and therefore can be implemented only in batch mode, while many real problems require on-line adaptation. In this paper we give the complete BPTT formulation for the IIR MLP, derive an already known on-line learning algorithm as a particular approximation of the BPTT, and propose a new approximated algorithm. Several computer simulations of identification of dynamical systems will also be presented to assess the performance of the approximated algorithms and to compare the IIR MLP with more traditional dynamic networks.

1. INTRODUCTION

Many practical problems, such as time series forecasting, non-linear filtering and control plant modelling, require non-linear dynamic models. Neural networks with internal temporal dynamic can be effectively used to perform these tasks.

In particular, the fully connected recurrent networks have been proposed and several learning algorithms have been developed for them, e.g. [1]. However, the complexity of such networks, together with the high computational and memory requirement during learning, can hinder them from being applied to several real problems. Therefore, less complex neural networks, based on the well known Multi Layer Perceptron (MLP), have been proposed. In these networks (here called dynamic MLP or DMLP), the temporal dynamic is modelled by using synapses with discrete-time internal memory [2], i.e. by replacing the classical synapsis either with a FIR filter [2] or with an IIR filter [2,3]. The DMLP with FIR or IIR synapses (here called FIR MLP and IIR MLP respectively) are of particular interest, since they can be considered as a non-linear extension of the classical linear FIR and IIR filters [9].

A general algorithm to adapt a dynamic neural network has been introduced in [4]. This algorithm, called Back-Propagation-Through-Time (BPTT), extends the classical backpropagation for memoryless networks to non-linear systems with memory. It is known that the BPTT is a non-causal algorithm [4], therefore it works only in batch mode, after the whole training set has been processed and stored, and requires a large amount of memory. Hence in many real problems the BPTT cannot be used to adapt the DMLP, but on-line learning algorithms are needed.

In order to develop on-line algorithms, some approximation to the BPTT must be made. Following this approach, several algorithms have been proposed for FIR-MLP: the causal version of the BPTT [5,12], the algorithm proposed in [6] (where the FIR-MLP was called Time-Delay-Neural-Network) and that presented in [2]. The derivations of these different algorithms as particular approximation of the BPTT are shown in [7].

On the contrary, only one on-line learning algorithm for IIR-MLP has been proposed by A.D.Back and A.C.Tsoi in [2] with a variant given in [8], without referring to the BPTT. In this paper we then give the complete BPTT formulation for the IIR-MLP, derive the already known algorithm [2,8] as a particular approximation of the BPTT, and propose a new approximated on-line algorithm which resembles the algorithm proposed in [5,12] for the FIR MLP.

Several computer simulations of identification of dynamical systems will also be presented to assess the performance of the approximated algorithms and to compare the IIR MLP with more traditional dynamic networks.

2. THE BPTT ALGORITHM FOR MLP WITH IIR SYNAPSES

It is well known that an IIR MLP contains a linear filter with poles and zeros (AR and MA part respectively) in each synapses. Due to the complexity of the resulting structure, a rigorous notation is needed.

Notation

In this paper, we use the following notation, which is a generalisation of that used in [7]:

M	number of layers in the network.
"l"	layer index. In particular l=0 and l=M denotes the input and output layer, respectively.
N_l	number of neurons of the l-th layer.
"n"	neuron index.
$x_n^{(l)}[t]$	output of the n-th neuron of the l-th layer. In particular n=0 refers to the bias inputs:

$x_0^{(l)}$=1. Note that $\{x_n^{(0)}\}$, n=1,...,N_0, is the input signal.

$L_{nm}^{(l)}$ length of the MA part of the synapsis of the n-th neuron of the l-th layer relative to the m-th output of the (l-1)-th layer. $L_{nm}^{(l)}\geq 1$ and $L_{n0}^{(l)}$=1.

$I_{nm}^{(l)}$ length of the AR part of the synapsis of the n-th neuron of the l-th layer relative to the m-th output of the (l-1)-th layer. $I_{n0}^{(l)}$=0.

$w_{nm(p)}^{(l)}$ (p=0,1, ..., $L_{nm}^{(l)}$-1) coefficients of the MA part of the synapsis. If $L_{nm}^{(l)}$=1, the synapsis has no MA part and the weight notation becomes $w_{nm}^{(l)}$. $w_{n0}^{(l)}$ is the bias.

$v_{nm(p)}^{(l)}$ (p=1, ..., $I_{nm}^{(l)}$) coefficients of the AR part of the synapsis. If $I_{nm}^{(l)}$=0 the synaptic filter is purely MA.

sgm(z) activation function.

sgm'(z) derivative of sgm(z).

$y_{nm}^{(l)}$[t] synaptic filter output at time t relative to the sinpasis of n-th neuron, l-th layer and m-th input. $y_{n0}^{(l)}=w_{n0}^{(l)}$.

$s_n^{(l)}$[t] "net" function relative to the n-th neuron of the l-th layer.

d_n[t] (n=1, ...,N_M) desired output sequence at time t.

As an example of such notation, a simple two layer (M=2) IIR MLP, with two inputs (N_0=2), one hidden neuron (N_1=1) with no MA and AR parts in each synapse ($L_{1m}^{(1)}$=1 and $I_{1m}^{(1)}=0$ for m=1, 2) and one output neuron (N_2=1) with both MA and AR parts in the synapses ($L_{11}^{(2)}$=3 e $I_{11}^{(2)}=2$), is shown in figure 1.

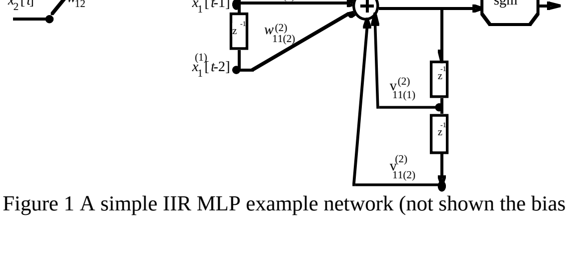

Figure 1 A simple IIR MLP example network (not shown the bias).

The forward phase

The forward phase at time t can be described by the following three equations evaluated for l=1,...,M and n=1,...,N_l:

$$y_{nm}^{(l)}[t]=\sum_{p=0}^{L_{nm}^{(l)}-1} w_{nm(p)}^{(l)} x_m^{(l-1)}[t-p]+\sum_{p=1}^{I_{nm}^{(l)}} v_{nm(p)}^{(l)} y_{nm}^{(l)}[t-p] \quad (1)$$

$$s_n^{(l)}[t]=\sum_{m=0}^{N_{l-1}} y_{nm}^{(l)}[t] \qquad x_n^{(l)}[t]=\mathrm{sgm}(s_n^{(l)}[t]) \quad (2)$$

The learning algorithm (BPTT)

The instantaneous global error at time t is defined as:

$$e^2[t]=\sum_{n=1}^{N_M} e_n^2[t] \qquad \text{with}$$

$$e_n[t]=d_n[t]-x_n^{(M)}[t]$$

so the global error over the training epoch is:

$$E^2=\sum_{t=1}^{T} e^2[t]$$

where T is the duration of the training epoch.
Let us define:

$$e_n^{(l)}[t]=-\frac{1}{2}\frac{\partial E^2}{\partial x_n^{(l)}[t]} \quad \text{and} \quad \delta_n^{(l)}[t]=-\frac{1}{2}\frac{\partial E^2}{\partial s_n^{(l)}[t]}$$

Therefore, using the chain rule:

$$\Delta w_{nm(p)}^{(l)}=-\frac{\mu}{2}\frac{\partial E^2}{\partial w_{nm(p)}^{(l)}}=-\frac{\mu}{2}\sum_{t=1}^{T}\frac{\partial E^2}{\partial s_n^{(l)}[t]}\frac{\partial s_n^{(l)}[t]}{\partial w_{nm(p)}^{(l)}}$$

that is possible to rewrite as

$$\Delta w_{nm(p)}^{(l)}=\sum_{t=1}^{T}\Delta w_{nm(p)}^{(l)}[t+1] \quad (3a)$$

where

$$\Delta w_{nm(p)}^{(l)}[t+1]=-\frac{\mu}{2}\frac{\partial E^2}{\partial s_n^{(l)}[t]}\frac{\partial s_n^{(l)}[t]}{\partial w_{nm(p)}^{(l)}}=$$

$$=\mu\,\delta_n^{(l)}[t]\frac{\partial s_n^{(l)}[t]}{\partial w_{nm(p)}^{(l)}} \quad (3b)$$

Similarly for the v weights we have:

$$\Delta v_{nm(p)}^{(l)}=\sum_{t=1}^{T}\Delta v_{nm(p)}^{(l)}[t+1] \quad (4a)$$

where

$$\Delta v_{nm(p)}^{(l)}[t+1]=\mu\,\delta_n^{(l)}[t]\frac{\partial s_n^{(l)}[t]}{\partial v_{nm(p)}^{(l)}} \quad (4b)$$

Now, the derivation of the BPTT learning algorithm for MLP with IIR synapses follows in a way similar to that used to find the BPTT algorithm for MLP with FIR synapses [5].
This case is obviously more difficult for the recursion that complicate the computation of the derivatives. The corresponding expressions in fact are:

$$\frac{\partial s_n^{(l)}[t]}{\partial w_{nm(p)}^{(l)}}=x_m^{(l-1)}[t-p]+\sum_{r=1}^{I_{nm}^{(l)}} v_{nm(r)}^{(l)}\frac{\partial s_n^{(l)}[t-r]}{\partial w_{nm(p)}^{(l)}} \quad (5a)$$

and

$$\frac{\partial s_n^{(l)}[t]}{\partial v_{nm(p)}^{(l)}}=y_{nm}^{(l)}[t-p]+\sum_{r=1}^{I_{nm}^{(l)}} v_{nm(r)}^{(l)}\frac{\partial s_n^{(l)}[t-r]}{\partial v_{nm(p)}^{(l)}} \quad (5b)$$

Note that such expressions are analogous to those found in the IIR linear adaptive filter theory [9, expressions 16a, 16b].

The last two formulas are exactly true only if the weights (w or v) are not time-dependent, or approximately true if they adapt slowly, i.e. the learning rate is sufficiently small [9]. In exact BPTT the weights update is performed only at the end of the learning epoch using the accumulated weight variations computed at every time instant, so that the above expressions are exact.

Using the chain rule it is possible to obtain:

$$e_n^{(l)}[k] = -\frac{1}{2}\frac{\partial E^2}{\partial x_n^{(l)}[k]} = \sum_{q=1}^{N_{l+1}}\sum_{t=1}^{T} -\frac{1}{2}\frac{\partial E^2}{\partial s_q^{(l+1)}[t]}\ \frac{\partial s_q^{(l+1)}[t]}{\partial x_n^{(l)}[k]}$$

By the last expression, under the hypothesis of IIR synaptic filter causality, the backpropagation through the layers can be derived:

$$e_n^{(l)}[t] = \begin{cases} e_n[t] & \text{for } l = M \\ \sum_{q=1}^{N_{l+1}}\sum_{p=0}^{T-t} \delta_q^{(l+1)}[t+p]\ \frac{\partial \mathrm{y}_{\mathrm{qn}}^{(l+1)}[t+p]}{\partial x_n^{(l)}[t]} & \text{for } l = (M-1),\ldots,1 \end{cases} \tag{6}$$

where the derivative is computed as (see expression (1)):

$$\frac{\partial \mathrm{y}_{\mathrm{qn}}^{(1+1)}[t+p]}{\partial x_n^{(l)}[t]} = \begin{cases} w_{qn(p)}^{(l+1)} & \text{if } 0 \le p \le L_{qn}^{(l+1)} - 1 \\ 0 & \text{otherwise} \end{cases} +$$

$$+ \sum_{r=1}^{\min(I_{qn}^{(l+1)},p)} v_{qn(r)}^{(l+1)} \frac{\partial y_{qn}^{(l+1)}[t+p-r]}{\partial x_n^{(l)}[t]} \tag{7}$$

and:

$$\delta_n^{(l)}[t] = e_n^{(l)}[t]\mathrm{sgm}'(s_n^{(l)}[t]) \tag{8}$$

with l=M,...,1 and n=1,...,N_l.

The expressions from (1) to (8) constitute the BPTT algorithm for IIR MLP. Note that if all the synapses contain only the MA part ($I_{nm}^{(l)}$=0 for each n,m and l), this algorithm reduces to the BPTT for the FIR MLP as in [5,7].

3. ON-LINE APPROXIMATED BPTT ALGORITHMS

As shown by expression (6), the exact BPTT algorithm is not causal, in fact the $e_n^{(l)}$ at time t depends on the $\delta_q^{(l+1)}$ quantities, taken at future time instants. Therefore the weights update can only be performed in batch mode, i.e. accumulating the weight variations at each time instant and using the exact formulas. However in many applications, especially in non-linear signal processing, an on-line learning algorithm is necessary, since it allows to track real-time problems where a learning epoch cannot be strictly defined. Another limitation of exact BPTT is that its memory requirement is proportional to the training epoch length (T). On line learning algorithms have instead a memory requirement that do not depend on T but only on the network architecture.

In the following, two on-line approximations are proposed and discussed.

Causal Backpropagation Through Time (CBPTT)

This approximation follows a similar idea found in the corresponding algorithm for FIR MLP [5,7,12]. This idea consists on introducing a suitable number of delays in the weight adaptation formulas in order to remove the non-causality. In other words:

$$weight^{(l)}[t+1] = weight^{(l)}[t] + weight\_\mathrm{var}iation^{(l)}[t+1-D_l]$$

where '$weight$' indicates either w or v and D_l is a suitable integer number. However, since the IIR synapses have an infinite memory due to the recursion, a truncation is necessary, so the truncated formula is:

$$e_n^{(l)}[t] = \begin{cases} e_n[t] & \text{for } l = M \\ \sum_{q=1}^{N_{l+1}}\sum_{p=0}^{Q_{l+1}} \delta_q^{(l+1)}[t+p]\ \frac{\partial \mathrm{y}_{\mathrm{qn}}^{(1+1)}[t+p]}{\partial x_n^{(l)}[t]} & \text{for } l = (M-1),\ldots,1 \end{cases} \tag{9}$$

where Q_{l+1} is appropriately chosen (usually at least greater than $\max\limits_{n,m}(L_{nm}^{(l+1)} - 1)$). It follows that

$$D_l = \begin{cases} 0 & \text{if } l = M \\ \sum_{i=l+1}^{M} Q_i & \text{if } 1 \le l < M \end{cases}$$

The causalised formula is then:

$$e_n^{(l)}[\tau - Q_{l+1}] = \sum_{q=1}^{N_{l+1}}\sum_{p=0}^{Q_{l+1}} \delta_q^{(l+1)}[\tau - p]\ \frac{\partial \mathrm{y}_{\mathrm{qn}}^{(1+1)}[\tau - p]}{\partial x_n^{(l)}[\tau - Q_{l+1}]} \tag{10}$$

for $l = (M-1),\ldots,1$

In the expressions (9) and (10), it has been supposed $\delta[t] = 0 \quad \text{if } t \notin [1,T]$ to simplify the formulas. The derivative in (9) and (10) is computed using (7).

The causalisation and the on-line update, compared to the batch mode case, is not a strong approximation if the learning rate is small enough, because in this case the weight variation is slow in relation to the delay D_l. The experiments show that if Q_{l+1} is large enough and the learning rate is small enough, the exact BPTT and the CBPTT give the same result.

Instantaneous Backpropagation (IBPTT)

This approximation follows the same idea of the corresponding algorithm for FIR MLP [2,7] and has been introduced in [2]. This simplification of the BPTT algorithm can be obtained by considering the global instantaneous squared error $e^2[t]$ instead of the total squared error E^2 . In this case the learning algorithm is simpler, since expression (6) is now replaced by

$$e_n^{(l)}[t] = \begin{cases} e_n[t] & \text{for } l = M \\ \sum_{q=1}^{N_{l+1}} \delta_q^{(l+1)}[t]\ w_{qn(0)}^{(l+1)} & \text{for } l = (M-1),\ldots,1 \end{cases} \tag{11}$$

In other words, now the errors are back-propagated only through the weights $w_{nm(0)}^{(l)}$ of each synapsis, i.e. the filters are not considered. This corresponds to the previous case when all the Q_l quantities are set to zero for each l (now D_l=0 for each l: no causalisation is needed). Note that the expression (11) results exactly the same as obtained by the IBP algorithm [7] for the FIR MLP.

4. EXPERIMENTAL RESULTS

In order to test the performance of the various algorithms with different networks (MLP with external dynamic, FIR MLP, IIR MLP), several experiments of identification of non-linear systems have been simulated.

The simplest neural network tested was a MLP with external dynamic (i.e. with external tapped delay lines for the input and the output feedback). The architecture of the network, the results of which are reported here is: two layers, eleven inputs (five of which are output feedback), five hidden neurons, and one output. The neurons of the hidden layer, as the output neuron, have a sigmoidal activation function. Standard static backpropagation was used to train this network, giving to the MLP the desired output and its delays as feedback inputs during the training phase (i.e. opening the feedback loop).

Then a FIR MLP with comparable architecture was tested. The DMLP architecture of the reported test is: two layers network with one input and one output, five hidden neurons that have the sigmoidal activation function, as the output neuron. The MA part of all the synaptic filters has five delays ($L_{nm}^{(l)}$=6 for each l, n, m).

Finally a IIR MLP was tried with the same architecture as the FIR MLP and (in the case reported here) the introduction of five AR delays in the hidden neurons synapses ($I_{nm}^{(1)}$=5 for each n, m), and four in the output neuron ($I_{nm}^{(2)}$=4 for each n, m). For all the architectures: sgm(z)=tanh(z/2). Very recently other simulations were made with architectures of the various networks chosen with the same number of free parameters obtaining about the same results and confirming the conclusions we state.

For the static MLP the standard and cumulative updating algorithms were tested, while for the FIR MLP and IIR MLP the BPTT, CBPTT, IBPTT algorithms were used. For the CBPTT algorithm of IIR MLP Q_2=10 was used for the results reported.

The results are given in terms of Mean-Square-Error (MSE), expressed in dB, computed on the learning set after each epoch (after all the input-output samples were presented) and averaged over 10 runs, each with a different weight initialisation.

The first set of experiments consisted in identifying the non-linear system with memory presented in [8]. The system is described by the following input-output relationship:

$$\begin{aligned} z(t) = {} & 0.0154x(t) + 0.0462x(t-1) + 0.0462x(t-2) + 0.0154x(t-3) + \\ & + 1.99z(t-1) - 1.572z(t-2) + 0.4583z(t-3) \\ y(t) = {} & \sin[z(t)] \end{aligned}$$

where $x[t]$ and $y[t]$ are the input and output signals at time t respectively.

A sequence $x[t]$ of 1000 points of random noise, with a uniform probability density function between -1 and +1, was generated. This sequence and the corresponding output sequence $y[t]$ constituted the training set for the various neural networks.

Figure 2 (top) shows the MSE of the various algorithms (Standard BP for the MLP, CBPTT and IBPTT for the FIR and IIR MLP) during the first 200,000 adaptation steps (200 epochs) and with an adaptation gain μ=0.005. It is evident that the IIR MLP performs much better than the MLP or FIR MLP in terms of accuracy (asymptotic MSE) and speed of adaptation. It is also clear that the CBPTT algorithm performs better than the IBPTT in terms of learning speed both for the FIR and IIR MLP. The BPTT for the IIR MLP is not reported because this learning rate is too high for it to be stable.

Figure 2 (bottom) shows the MSE of: Standard BP for the MLP, CBPTT for the FIR MLP and BPTT, CBPTT, IBPTT for the IIR MLP during the first 300,000 adaptation steps (300 epochs) and with an adaptation gain μ=0.0005. This small learning rate has been chosen to show that the exact BPTT and the causalised one became equivalent for small μ (see the plots).

The second set of experiments was carried out on the more realistic problem of identifying a baseband equivalent PAM transmission system in presence of a non linearity [10]. The pulse shaping circuit transforms the discrete-time symbols stream $a[n]$ in a continuous-time signal $v(t)$ (PAM) by a filter with a raised-cosine shape and roll-off factor α . The signal $v[t]$ is then processed by the High-Power-Amplifier (HPA) which is modelled here by the following input-output relationship:

$$w[t] = \frac{2\ v[t]}{1 + v^2[t]}.$$

The peak power of the input signal $v[t]$ is set to the value of β dB (back-off factor), being β=0 dB the normalised unit power. The HPA output $w[t]$ is corrupted by an additive white Gaussian noise $z[t]$, producing the final signal $y[t]$ with a given Signal-To-Noise ratio (SNR). The overall system is clearly dynamic and non-linear.

A neural network approach to equalise this system has already been proposed in the technical literature (see for example [11]). In our experiment a neural network was used instead to identify a sampled version of the system. For this purpose, $\{a[n]\}$ was chosen to be a random sequence of 512 symbols drawn from a 16-symbols alphabet. The pulse shaping filter had a roll-off α=0.3, and the HPA back-off β was set to -2 dB. The noise level was very low: SNR=80 dB.

The networks were selected with the same characteristics of those used in the first set of experiments.

By using an over sampling ratio of four at the output with respect to the symbol rate, the sequences $\{a[n]\}$ of 512 symbols and $\{y[t]\}$ of 2048 samples were used as the learning set and again the MSE was computed

after all the 512 input symbols (epoch) were presented.
The MSE is plotted in figure 3 vs. the training epoch. In particular figure 3 shows performance of the training algorithms (Standard BP for the MLP, CBPTT for the FIR MLP and CBPTT, IBPTT for the IIR MLP) during the first 204.800 adaptation steps (100 epochs) with μ=0.005. Again it is evident that the IIR MLP performs much better than the MLP or FIR MLP in terms of accuracy and speed of adaptation. It is also clear that the CBPTT algorithm performs better than the IBPTT in terms of learning speed for the IIR MLP.
Again the BPTT for the IIR MLP is not reported because this learning rate is too high for it to be stable.

CONCLUSIONS

The results presented in this work show that the MLP with IIR synapses has superior modelling capabilities (accuracy) with respect to more traditional networks, namely MLP with external dynamic and FIR MLP (or TDNN), for the identification of the systems tested. For this reason we developed a general formulation to derive on-line algorithms based on BPTT and proposed a new learning method, that, as a limit case, gives the one already known in literature. The proposed algorithm has better stability and higher speed of convergence with respect to the previous one, as expected by the theoretical development and confirmed by simulations. Stability and speed of convergence are very important in real on-line applications, where time varying systems have to be tracked.
The only drawback of the algorithm is a slight increase in complexity, that however can be reduced since even small values of the truncation parameter (i.e. low complexity) can attain much better results than the already known algorithm, as it will be shown in future works.

REFERENCES

[1] R.J. Williams, J. Peng, "*An efficient gradient-based algorithm for on line training of recurrent network trajectories*", Neural Computation 2: 490-501, 1990.

[2] A.D. Back, A.C. Tsoi, "*FIR and IIR synapses, a new neural network architecture for time series modelling*", Neural Computation 3: 375-385, 1991.

[3] A.C. Tsoi, A.D. Back, "*Locally recurrent globally feedforward networks: a critical review of architectures*", IEEE Transactions on Neural Networks, vol. 5, no. 2, 229-239, March 1994.

[4] P.J. Werbos, "*Backpropagation through time: what it does and how to do it*", Proceedings of IEEE, Special issue on neural networks, vol. 78, No. 10, pp.1550-1560, October 1990.

[5] E.A. Wan, "*Temporal backpropagation for FIR neural networks*", Proceedings of the International Joint Conference on Neural Networks, 1:575-580, 1990.

[6] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K.J. Lang, "*Phoneme recognition using time-delay neural networks*", IEEE Trans. on Acoustic, Speech, and Signal Processing, Vol. 37, No.3, March 1989.

[7] N. Benvenuto, F. Piazza, A. Uncini, "*Comparison of four learning algorithms for multilayer perceptron with FIR synapses*", Proceeding of the IEEE International Conference of Neural Network, 1994.

[8] A.D. Back, A.C. Tsoi, "*A simplified gradient algorithm for IIR synapse Multilayer Perceptron*", Neural Comp. 5: 456-462, 1993.

[9] J.J. Shynk, "*Adaptive IIR filtering*", IEEE ASSP Magazine, April 1989.

[10] J.G. Proakis, "*Digital Communications*", McGraw-Hill, 1989.

[11] S. Chen, G.J. Gibson, C.F.N. Cowan and P.M. Grant, "*Adaptive equalisation of finite non-linear channels using multilayer perceptrons*", Signal Processing, vol. 20, pp. 107-119, June 1990.

[12] S. Haykin, "*Neural Networks: a comprehensive foundation*", IEEE Press-Macmillan, pp 506-515, 1994.

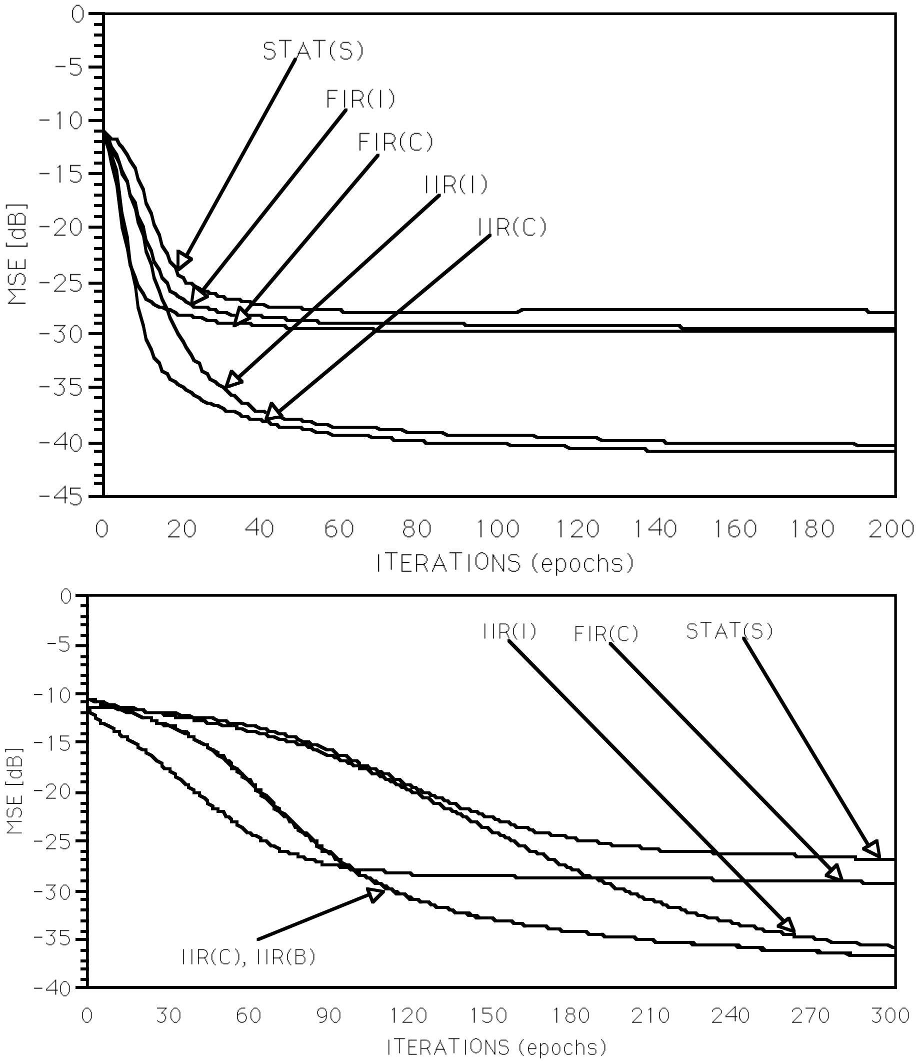

Figure 2 Convergence performance of various algorithms and networks on identifying a non linear system with internal memory. Within the parenthesis the first letter of the used algorithm. μ=0.005 (top), μ=0.0005 (bottom).

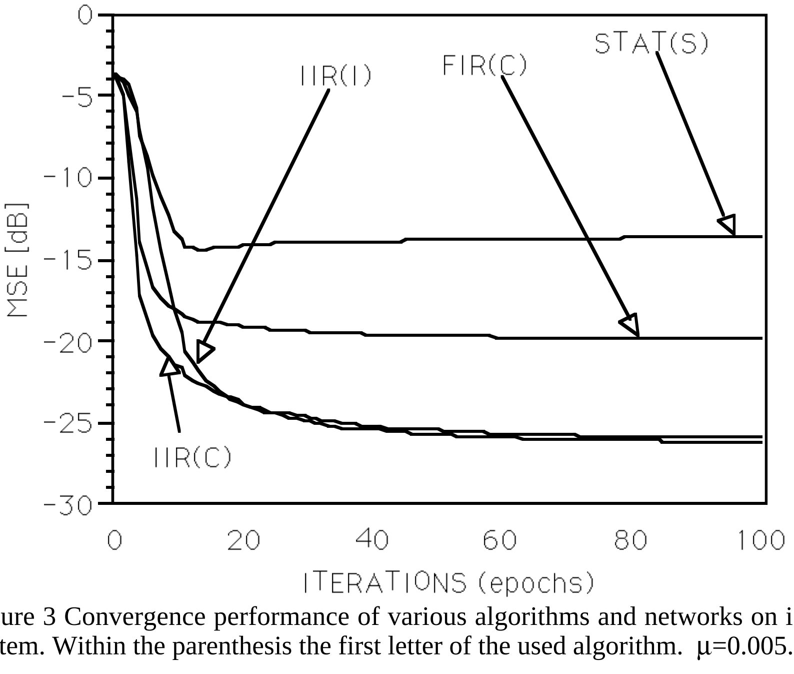

Figure 3 Convergence performance of various algorithms and networks on identifying a 16-PAM transmission system. Within the parenthesis the first letter of the used algorithm. μ=0.005.