A NEW IIR-MLP LEARNING ALGORITHM FOR ON-LINE SIGNAL PROCESSING

Paolo Campolucci[*], *Simone Fiori, Aurelio Uncini, Francesco Piazza* [+]

Dipartimento di Elettronica e Automatica
Università di Ancona, 60131 Ancona, Italia.
[*]e-mail: paolo@eealab.unian.it

ABSTRACT

In this paper we propose a new learning algorithm for locally recurrent neural networks, called Truncated Recursive Back Propagation which can be easily implemented on-line with good performance. Moreover it generalises the algorithm proposed by Waibel et al. for TDNN, and includes the Back and Tsoi algorithm as well as BPS and standard on-line Back Propagation as particular cases. The proposed algorithm has a memory and computational complexity that can be adjusted by a careful choice of two parameters *h* and *h*' and so it is more flexible than a previous algorithm by us.
Although for the sake of brevity we present the new algorithm only for IIR-MLP networks, it can be applied also to any locally recurrent neural network.
Some computer simulations of dynamical system identification tests, reported in literature, are also presented to assess the performance of the proposed algorithm applied to the IIR-MLP.

1. INTRODUCTION

An increasing number of applications of dynamic recurrent neural networks has been developed for Digital Signal Processing (DSP) [1,2].
Recurrent neural networks can provide better modeling accuracy compared to buffered static Multi Layer Perceptron (MLP) or MLP with Finite Impulse Response (FIR) filter synapses [3], often known as Time Delay Neural Network (TDNN) [4] even if these networks are also used for simplicity. Feedbacks are necessary when a long and complex temporal dynamics is required. Fully recurrent networks are general but difficult to train [5]. Especially for DSP problems, for which, in the case of stability, a forgetting behaviour [6] is usually required, the MLP with Infinite Impulse Response filter synapses (IIR-MLP) [3,5] can exibit better capabilities, due to the prewired forgetting behaviour (typical of locally recurrent networks). In fact IIR-MLP can be considered as a non-linear extension of the linear adaptive IIR filter.
Similar architectures called Local Feedback Multi-Layered Networks (LF-MLN) were proposed by P. Frasconi, M. Gori and G. Soda [6] for speech recognition, and a specific learning algorithm (Back Propagation for Sequences or BPS) was derived.
As far as the learning is concerned, in [7] a general algorithm to adapt dynamic neural networks has been introduced. 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 [7], therefore it works only in batch mode, after the whole signal 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 network, but on-line learning algorithms are needed.
In order to develop on-line algorithms there are two classical choices: approximating BPTT [8] or using the computationally very expensive Real Time Recurrent Learning (RTRL) [9].
In this paper we propose a new learning algorithm, called Truncated Recursive Back Propagation (TRBP) which can be easily implemented on-line with good performance. Moreover it generalises the algorithm proposed by Waibel et al. in [4] for TDNN, and includes the Back and Tsoi algorithm [3] as well as BPS [6] and standard on-line Back Propagation as particular cases. The proposed algorithm has a memory and computational complexity that can be adjusted by a careful choice of two parameters *h* and *h*' and so it is more flexible than the algorithm presented in [10,11] with similar performance.
Although for the sake of brevity we present the new algorithm only for IIR-MLP networks, it can be applied also to any locally recurrent neural network [5].
Some computer simulations of dynamical system identification tests, reported in literature [12], will also be presented to assess the performance of the proposed algorithm applied to the IIR-MLP.

[+]This research was supported by the Italian MURST.
Please send comments and suggestions to the first author.

2. THE BATCH MODE ALGORITHM (RBP) AND ITS ON-LINE VERSION (TRBP)

The Recursive Back-Propagation (RBP) batch mode learning algorithm is described by the following expressions with the same notation introduced in [10] (an extension of Widrow's one) and here summarised:
N_l is the number of neurons in layer l, M is the number of layers, T is the duration of the learning epoch, $y_{qn}^{(l)}$ is the output of the IIR synaptic filter of the neuron q in layer l and input n, $x_n^{(l)}$ is the output of the neuron n in layer l, $s_n^{(l)}$ the related *net*, $L_{nm}^{(l)}$ -1 the order of the Moving Average (MA) part of the corresponding IIR synaptic filter, $I_{nm}^{(l)}$ the analogous for the Auto Regressive (AR) part, $d_n[t]$ is the n-th desired output signal, μ is the learning rate, $E(t_0,t_1)$ the squared error in the time range $[t_0,t_1]$, $e_n^{(l)}[t] = -\frac{1}{2}\frac{\partial E(1,T)}{\partial x_n^{(l)}[t]}$ is the *backpropagating error*, $\delta_n^{(l)}[t] = -\frac{1}{2}\frac{\partial E(1,T)}{\partial s_n^{(l)}[t]}$ is the usual *delta*, $weight_{nm(p)}^{(l)}$ indicates either a numerator ('*w*') or denominator ('*v*') coefficient of the IIR filter rational function of neuron n, input m, layer l and delay p, sgm(.) is the activation function.
The forward phase at time t can be described by the following three equations evaluated for l=1,...,M and n=1,...,N_l:

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

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

The weights variations are computed, by chain rule, as:

$$\Delta weight_{nm(p)}^{(l)} = -\frac{\mu}{2}\frac{\partial E(1,T)}{\partial weight_{nm(p)}^{(l)}} = \sum_{t=1}^{T} \Delta weight_{nm(p)}^{(l)}[t+1] \quad (3)$$

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

where

$$\delta_n^{(l)}[t] = e_n^{(l)}[t]\,\mathrm{sgm}'(s_n^{(l)}[t]) \quad (5)$$

$$\frac{\partial s_n^{(l)}[t]}{\partial w_{nm(p)}^{(l)}} = x_m^{(l-1)}[t-p] + \sum_{r=1}^{I_{nm}^{(l)}} v_{nm(r)}^{(l)} \frac{\partial s_n^{(l)}[t-r]}{\partial w_{nm(p)}^{(l)}} \quad (6)$$

for the weights of the MA part, and

$$\frac{\partial s_n^{(l)}[t]}{\partial v_{nm(p)}^{(l)}} = y_{nm}^{(l)}[t-p] + \sum_{r=1}^{I_{nm}^{(l)}} v_{nm(r)}^{(l)} \frac{\partial s_n^{(l)}[t-r]}{\partial v_{nm(p)}^{(l)}} \quad (7)$$

for the weights of the AR part.
Under the hypothesis of IIR synaptic filter causality, it holds true, by chain rule:

$$e_n^{(l)}[t] = \begin{cases} d_n[t] - x_n^{(M)}[t] & \text{for } l = M \\ \sum_{q=1}^{N_{l+1}} \sum_{p=0}^{T-t} \delta_q^{(l+1)}[t+p]\, \frac{\partial y_{qn}^{(l+1)}[t+p]}{\partial x_n^{(l)}[t]} & \text{for } l = (M-1),\ldots,1 \end{cases} \quad (8)$$

$$\frac{\partial y_{qn}^{(l+1)}[t+p]}{\partial x_n^{(l)}[t]} = \begin{cases} w_{qn(p)}^{(l+1)} \text{ if } 0 \le p \le L_{qn}^{(l+1)} - 1 \\ 0 \quad \text{otherwise} \end{cases} + \sum_{r=1}^{\min(I_{qn}^{(l+1)},\, p)} v_{qn(r)}^{(l+1)} \frac{\partial y_{qn}^{(l+1)}[t+p-r]}{\partial x_n^{(l)}[t]} \quad (9)$$

for layer l=M,...,1 and neuron n=1,...,N_l. Note that expressions (6) (7) and (9) are known in the linear adaptive IIR filter theory [13], the only difference is that some indexes are present to specify the synapsis in the network where the IIR filter is.
It is easy to see that this RBP algorithm, as BPTT, is not causal, in fact $e_n^{(l)}$ at time t depends on $\delta_q^{(l+1)}$ at future time instants. 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 of non-linear signal processing, an on-line learning algorithm is necessary. Therefore we want to derive a new method to approximate this batch mode algorithm whose inspiration comes from the algorithm proposed by Williams and Peng for fully recurrent neural networks [8].
Must be stressed that the RBP algorithm is not a version of BPTT [7,8] nor one of RTRL [9] even if in the first papers [10,11] we used the name BPTT. However, for the first time, it implements some features of both algorithms: the backward error propagation of BPTT is used in expressions (3), (5) and (8); the recursive forward derivatives calculation commonly used in the RTRL and output-error approach (Recursive Prediction Error or RPE) [13] is implemented in formulas (6) and (7); expressions (4) and (9) have no direct link to BPTT nor to RTRL. This means that the new RBP algorithm applied to the IIR-MLP implements the output-error RPE approach for the IIR filters adaptation and the BPTT error backpropagation to exploit the neural layered structure; the resulting method allows an accurate and efficient on-line gradient computation in locally recurrent neural networks.
First the RBP algorithm must be causalized. This is provided by choosing the squared error computed at the current time instant (τ in the following) and not over all the sequence, so that the gradient descent can be expressed by:

$$\Delta weight_{nm(p)}^{(l)} = -\frac{\mu}{2}\frac{\partial E(\tau,\tau)}{\partial weight_{nm(p)}^{(l)}} = \sum_{t=1}^{\tau} \Delta weight_{nm(p)}^{(l)}[t+1] \quad (10)$$

Causalization can be obtained simply substituting the final instant T of the sequence with the current instant τ in (3) and in (8). In this way the upper index of the inner summation in (8) is the current time minus the index t of

the summation in (10). So every time step, expression (8) is evaluated for t=1 to τ using $\delta_q^{(l+1)}[.]$ up to time τ for each time instant, so that the calculation is causal.
Since this implementation would require a memory and computational complexity for each iteration that grows with time, a forgetting mechanism must be implemented for on-line training. Forgetting the old past history is very reasonable and recommended by many authors, e.g. [8]. The obtained performance are competitive [8] even with the RTRL algorithm that computes a slightly approximated gradient of the instantaneous error [9]. The only expression that must be changed is (10) that becomes:

$$\Delta weight_{nm(p)}^{(l)} = \sum_{t=\tau-h+1}^{\tau} \Delta weight_{nm(p)}^{(l)}[t+1] \qquad (11)$$

where h is the length of the considered history buffer .
Now the algorithm can be implemented on-line with parameters changes computed by (11), performed every time step. It can be shown that since we are considering instantaneus error cost function, the error injection [8] should be performed only for the current time step so that in this case, for the last layer, $e_n^{(l)}[t]$ is zero for $t<\tau$; this simplifyies the calculation of expression (11) for the last layer since the corresponding *delta* is also zero and so only one term remains in (11) when l=M. For the same reason, expression (8) can be easily computed for l=M-1 since again only one term remains in the inner summation.
To simplify the algorithm a modification can be implemented: updating the coefficients every h' instants instead of every instant. This reduces the computational complexity basically by a factor of h'; if h-h' is large enough (h>=h' always) the approximation gives good performance [8]. However when applications that require adaptation every time instant are involved, (e.g. on-line tracking of fast varying systems), h' can be chosen equal to one, of course.
For the general method with h'>1, the error injection must be performed for the time indexes t from τ-h'+1 to τ so that more terms are present in the summations in (8) and (11) and not only one as previously explained. This correspond to compute the gradient of $E(\tau\text{-}h'\text{+}1,\tau)$.
Even when h'=1, if a cost function smoother than the instantaneous error is desired such as $E(\tau\text{-}N_C\text{+}1,\tau)$ (with N_C appropriately chosen and constrained to be $N_C \geq h'$), the error injection can be performed for all $t \in [\tau - N_c + 1, \tau]$.
In this paper we are assuming N_C=1, but the extension is trivial.
We will call the algorithm defined in this way Truncated Recursive Back Propagation (TRBP). From now on the algorithm in [10,11] will be called Causal RBP (CRBP).
To improve the accuracy, when IIR-MLP or locally recurrent layered networks are considered, it can be useful to make h depending on the layer l in (11) and chooses it approximately equal to one plus the summation of the maximum memory of the synaptic filters (roughly estimated) from layer l+1 to the last one, i.e.:

$$h_l = \begin{cases} h' & \text{if } l = M \\ 1 + \sum_{i=l+1}^{M} Q_i & \text{if } 1 \leq l < M \end{cases} \qquad (12)$$

where Q_i is an estimate of the maximum memory for the filters in layer i. This choice is reasonable since the truncation of (10) depends on the value $\delta_n^{(l)}[t]$ varying t that in turn depends on the memory of the synaptic filters of layers from l+1 to M.
It is possible to give a formal condition under which the truncation of (10) is feasible:

$$\lim_{h\to\infty} \delta_n^{(l)}[\tau - h] = \lim_{h\to\infty} -\frac{1}{2}\frac{\partial E(\tau,\tau)}{\partial s_n^{(l)}[\tau - h]} = 0 \text{ for each } n, \tau$$

or equivalently $\lim_{h\to\infty} \frac{\partial x_k^{(M)}[\tau]}{\partial s_n^{(l)}[\tau - h]} = 0$ for each k, n, τ.

This condition is satisfied for IIR-MLP if all the IIR filters of an arbitrary chosen layer from the l+1-th to the last one are stable or more in general for locally recurrent multilayered neworks if all the neurons in an arbitrary layer after the l-th exhibit forgetting behaviour [6] (the proof can be given by the chain rule). This condition is different from that which allows the truncation needed by CRBP (in formula (8)) since in that case the stability of the filters of all layers after the first was needed. Moreover only for the sake of truncation of (10), the previous condition can be relaxed to the following: at least one branch in every path from one arbitrary neuron of layer l-th to a neuron of the last layer must be a stable IIR filter. Anyway for the overall TRBP algorithm all the synaptic IIR filters in the network are required to be stable, but the previous analysis seem to indicate a less sensibility to truncation for TRBP than for CRBP if M>2.
A difference with the approximation proposed in [10,11] is that more terms than the current time coefficient variations are computed and accumulated for each iteration, potentially with a better approximation of the true gradient. The new method is simpler than the previous one if h/h' is chosen reasonably close to one (with the limitation that h-h' should be large enough for accuracy). A good choice can be h=2h' with h' large enough.
It is interesting to note that TRBP with the previous choice for h_l using Q_i equal to the maximum FIR filter order of layer i-th and h'=1 gives as particular case the algorithm proposed by Waibel et al. in [4] for MLP with FIR synapses (or TDNN). It has a nice geometric interpretation since it is obtained applying static BP to the network unfolded in time replicating network substructure for each delay. The resulting network has no internal memory but the inputs are the original ones and their delayed versions. Our formulation is much more general since it allows feedbacks in the network.
Moreover TRBP with h=h'=1 gives the Back and Tsoi algorithm [3] for IIR-MLP. Of course, if all the synaptic filters have no memory (the IIR-MLP becomes a standard MLP) and h'=1 TRBP particularises to standard on-line BP

(h_l=1 for each l according to (12)). Even BPS for LF-MLN [6] is a particular case of TRBP.

3. SIMULATIONS

To asses the performance of the new learning algorithm and compare it to the previous ones we chose two difficult non-linear dynamic systems identification tests from literature [12], for which IIR-MLP or locally recurrent neural networks give better modelling results than buffered MLP, FIR-MLP and fully recurrent neural networks.
The first system is a first-order single-input single-output system described by state variable and with non-linearity and memory which are non-separable. The second system is a second-order single-input two outputs system again with non-linearity and memory which are not separable.
For the system identification a 1000 samples uniform random noise sequence in the range [-1.25,1.25] was used.
The simulations reported refer to a IIR-MLP network with 3 sigmoidal neurons in the hidden layer and 1 or 2 linear output neurons respectively for the first and second system. For the first system the IIR-MLP used has MA order equal to 4 and AR order equal to 2 whereas for the second system the orders are respectively 3 and 1. The learning rate is 0.01.
Results for TRBP with h=4, h'=1, CRBP with Q_2=2 and Back-Tsoi algorithm for training an IIR-MLP are reported in Figure 1 for the first system and Figure 2 for the second one, in terms of Mean Square Error (MSE) during training. The simulations show the good performance of the proposed learning algorithm.

REFERENCES

[1] S. Haykin, "*Neural Networks Expand Sp's Horizons*" IEEE Signal Processing Magazine, vol.13 no. 2, March 1996.

[2] S. Haykin, "*Neural Networks: a comprehensive foundation*", IEEE Press-Macmillan, 1994.

[3] 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.

[4] 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.

[5] 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.

[6] P.Frasconi, M.Gori, G.Soda, "*Local Feedback Multilayered Networks*", Neural Computation 4: 120-130, 1992.

[7] 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.

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

[9] R.J. Williams, D. Zipser, "*A Learning Algorithm for Continually Running Fully Recurrent Neural Networks*", Neural Computation 1: 270-280, 1989.

[10] P. Campolucci, F. Piazza, A. Uncini, "*On-line learning algorithms for neural networks with IIR synapses*", Proc. of the IEEE International Conference of Neural Networks, Perth, Nov. 1995.

[11] P. Campolucci, A. Uncini, F. Piazza, "*Causal Back Propagation Through Time for Locally Recurrent Neural Networks*", IEEE International Symposium on Circuits and Systems, Atlanta, May 1996.

[12] D. Obradovic, "*On-line Training of Recurrent Neural Networks with Continuos Topology Adaptation*", IEEE Trans. on Neural Networks, vol.7, no.1, January 1996.

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

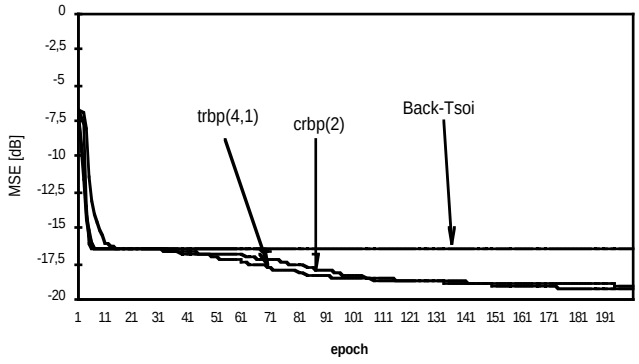

Figure 1. First system identification test results

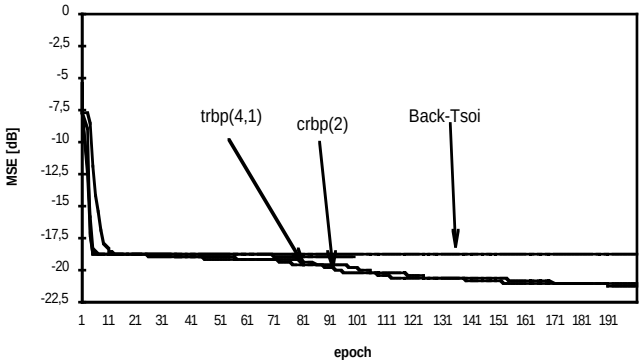

Figure 2. Second system identification test results