Real Time System Modelling Using Locally Recurrent Neural Networks

Paolo Campolucci, Aurelio Uncini, Francesco Piazza
Dipartimento di Elettronica e Automatica - Università di Ancona
via Brecce Bianche, 60131 Ancona, Italy
Tel. +39 (71) 220 4453 Fax +39 (71) 220 4464, E-mail: paolo@eealab.unian.it

Abstract - In this paper dynamic neural networks for system modelling are considered: architectural issues are presented but the paper focuses on learning algorithms that work real-time. A recent architecture called locally recurrent neural network is presented in its different versions and compared to traditional networks internally static but provided with external buffer and MLP with finite memory synapses. Simulations results show better modelling performance for locally recurrent networks and so an improved training algorithm is developed for them: Causal Back Propagation Through Time.

Validation tests shows that the networks are modelling the underlying system and not just overfitting the data.

I. INTRODUCTION

Neural networks with temporal dynamic are able to perform systems modelling due to their non-linear dynamic behavior; moreover they can be adapted recursively while the system (plant) is running. This makes dynamic neural networks very attractive for system modelling.

Neural networks with internal memory can provide better matching accuracy compared to static buffered neural networks (e.g. Time Delay Neural Network, TDNN [13]) even if this approach is often used for simplicity. In particular Infinite Impulse Response (IIR) MLP, see fig. 1 and 2, has better potential capabilities than TDNN or MLP with Finite Impulse Response (FIR) synapses [13]. Moreover TDNN without feedback and FIR MLP are particular case of IIR MLP.

Another architectural choice is fully recurrent networks [13]: they are general but difficult to train [2], so, especially for identification problems, the IIR MLPs can have greater capabilities, due to the nice properties of the IIR filter.

Similar architectures called Local Feedback MultiLayered Networks (LF MLN) were proposed by P.Frasconi, M.Gori and G.Soda [10,11]: the activation feedback MLN (fig. 3) and the output feedback MLN (fig. 4); they differ only on where the feedback is taken in the neuron model.

In this paper the learning algorithm proposed will be explained referring to IIR MLP only, for space limitation, but it is not difficult to modify the derivation for LF MLP. Instead the results of LF MLN modelling simulations will be reported too for comparison purpose.

The LF MLN here considered are a generalization of the original ones since we allow to have dynamic neurons in every layer, instead in [10,11] they are only permitted in the first layer, while the others layers must have static units. This constraint was imposed in order to simplify the learning algorithm, but we believe that permitting communications between dynamic units can give a better modelling.

To adapt locally recurrent neural networks two algorithms were proposed in the past: one by A.D. Back and A.C. Tsoi [1] (here called instantaneous backpropagation through time or IBPTT) with a variant given in [7] and the other by Gori et al. [10] called Back Propagation for Sequences (BPS).

Instead a general algorithm to adapt a dynamic neural network has been introduced in [3]. 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 [3], 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 network, but on-line learning algorithms are needed.

In order to develop on-line algorithms, one approach is to make some approximation to the BPTT. This approach is general and flexible: in this way previous learning algorithms can be derived as particular case.

In this paper we also discuss a new on-line algorithm: Causal-Back-Propagation-Through-Time, CBPTT, which is particularly useful for real-time system modelling by dynamic neural networks. The new algorithm includes the Back and Tsoi algorithm as well as BPS and Wan's Temporal Back Propagation as particular case. The proposed algorithm is less complex than those based on the unfolding in time approach following Williams-Peng approach [12,5,6], and is faster and more stable than IBPTT.

Many systems modelling results are provided to assess the capabilities of locally recurrent networks trained by CBPTT.

II. REAL TIME LEARNING ALGORITHM

The exact BPTT algorithm for the IIR MLP is described by the following expressions [9]:

\[ \Delta w_{ext}(t+1) = \sum_{i=1}^{T} \Delta w_{ext}(i)[t+1] \]  
\[ \Delta w_{int}(t+1) = \mu \delta(t) \frac{\partial \delta(t)}{\partial w_{int}(t+1)} \]  

where 'weight' indicates either a numerator ('w') or denominator ('v') coefficient of the IIR filters rational function, \( \mu \) is the learning rate, and:

\[ \frac{\partial \delta(t)}{\partial w_{int}(t+1)} = \sum_{i=1}^{T} \delta(t-p) + \sum_{i=1}^{T} \delta(t-r) \]  

for the weights of the moving average (MA) part, and:

\[ \frac{\partial \delta(t)}{\partial w_{int}(t+1)} = \sum_{i=1}^{T} \delta(t-p) + \sum_{i=1}^{T} \delta(t-r) \]  

for the weights of the auto regressive (AR) part.

Under the hypothesis of IIR synaptic filter causality, it holds true:
The causalized formula is then:

\[ e^{(l)}_{\text{in}}(\tau - Q_{\text{in}}) = \sum_{p=0}^{N_q} \delta^{(l)}_{\text{syn}}(\tau - p) \frac{\partial y^{(l)}_{\text{out}}}{\partial x^{(l)}_{\text{in}}}(\tau - Q_{\text{in}}) \]

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

In the expression (9), it has been supposed \( \delta^{(l)} = 0 \) if \( i \notin [1,T] \) to simplify the formula and \( \tau = t + Q_{\text{in}} \). The derivatives in (9) are computed recursively as before, so with (6). The causalization and the on-line update, compared to the batch mode case, are 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 \( D_l \).

It is also possible to justify the truncation approximation with the following theorem:

**Theorem 1**: If a linear, time invariant IIR filter is asymptotically stable (impulse response goes to 0 if \( t \to \infty \), i.e. all the poles are inside the unit circle) then

\[ y_{\text{out}}(t) \to 0 \quad \text{as} \quad p \to \infty \]

where \( y_{\text{out}}(t) \) is the output of the filter and \( x(t) \) the input, at time \( t \). The proof can be done considering that in (6) (it holds for general IIR filter) the recursion coefficients are the same of the corresponding IIR filter so the poles are the same and the asymptotic properties of that derivative are the same as for the filter output (i.e. respectively for \( p \to \infty \) and \( t \to \infty \)).

So in the reasonable hypothesis that the IIR filters are asymptotically stable (we verified this condition in simulations) then with \( p \) large enough, the contributions to the summation in \( p \) in expression (5) become negligible, making truncation possible. The experiments show that if \( Q_{\text{in}} + 1 \) is large and the learning rate is small enough, the exact BPTT and the CMBPTT give the same results.

The on-line algorithm proposed by Back and Tsoi [1,7] that is the only learning algorithm proposed for FIR MLP, is a particular case of our approximation if a strong truncation of the summation is used: \( Q_{\text{in}} + 1 = 0 \) for each \( l \). In this way the backpropagation is considering only the instantaneous influence of the IIR filter input to the output (the coefficient \( w^{(l)}_{\text{in}}(0) \)). No causality is needed (being \( D_{\text{in}} = 0 \) for each \( l \)) and the algorithm is very simple. However with only few additional memory terms in the backpropagation \( Q_{\text{in}} + 1 > 0 \) is possible to reach much better stability and speed of convergence, due to the introduction of recursion on the calculation.

**III. SYSTEM MODELLING SIMULATIONS RESULTS**

Many simulations were performed on the three locally recurrent architectures (fig.2,3,4) while for comparison two traditional neural networks were also tested, namely the internally static MLP (i.e. static MLP with input and/or output buffer) and the FIR MLP. The results reported in this paper refer to two tests of identification of non-linear dynamical systems.

The number of delays for the five architectures (i.e. buffer length for the MLP or MA-AR order for the FIR or IIR MLP) was chosen to have the best performance (approximately) from each of the five networks, but the total number of free parameters was fixed (40 parameters, bias included).

Common features of all the networks are: two layers, three hidden neurons with hyperbolic tangent activation function, one linear output neuron. Three different
learning algorithms were used: standard static backpropagation (open loop approximation) for MLP, temporal backpropagation for FIR MLP [4] and the proposed algorithm for IIR MLP; to simplify the comparison, momentum term and adaptive learning rate were not implemented even if they could improve performance. 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 initialization.

The first set of experiments consisted in identifying the non-linear system with memory presented in [7]. From fig. 5 it is evident that the locally recurrent MLPs performs much better than the static MLP or FIR MLP in terms of accuracy (asymptotic MSE) and speed of adaptation. Figure 7 shows the performance of the CBPPTT learning algorithm with different values of the truncation parameters $Q_2$ ($Q_2=0$ is the IBPPTT algorithm). It is clear that the CBPPTT algorithm performs much better than the previous IBPPTT even with small $Q_2$, i.e. with a small number of recursive terms.

The second set of experiments was carried out on the more realistic problem of identifying a baseband equivalent PAM transmission system in presence of non-linearity [6]. A pulse shaping block 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 $\alpha$. This is the dynamic part of the system. The signal $v(t)$ is then processed by the High-Power-Amplifier (HPA) which is non linear. See [6,9] for details of the communication system. Figure 6 shows the performance of the five neural architectures: again the locally recurrent MLPs perform much better than the two conventional MLPs. Again for this test, CBPPTT is the algorithm of choice, see fig. 8. Generalization tests for the identification of the 16 PAM channel were made using different realizations of the symbols stream in input, results are in table 1 e 2.

IV. CONCLUSIONS

Since the testing error for all the neural networks trained is close to the learning error, and very low, we know that these networks have learned the problem and not just the data, i.e. the trained model has the same temporal structure of the communication system (plant). This fact proves that dynamic neural networks can be used for system modelling; since the best performance are obtained with locally recurrent neural networks we proposed a learning algorithm to exploit their capabilities. In fact since this neural architectures have internal dynamic they require new learning algorithm not just simple variations of Back-Propagation.

The simulations suggests for this problems the use of IIR MLP or locally recurrent activation feedback MLN: they have almost the same performance; output feedback MLN is not as good as the previous ones but much better than TDNN or FIR MLP that are equivalent.

The proposed learning algorithm has better stability and higher speed of convergence with respect to the previous one (IBPPTT) and the computational complexity is only slightly higher. CBPPTT can be derived for each dynamic neural network because it is a general approach to transform BPTT in an on-line algorithm and so it can be used every time BPTT is used allowing on-line applications. The algorithm here proposed, CBPPTT, particularizes to Wan's for FIR MLP.

This work is supported by the Ministero dell'Università e della Ricerca Scientifica e Tecnologica of Italy.

REFERENCES


Fig. 3 Model of the neuron for the locally recurrent, activation feedback MLP.

Fig. 4 Model of the neuron for the locally recurrent, output feedback MLP.

Fig. 5 Comparison among different architectures performed on the identification of the non-linear dynamic system in (7). STAT is internally static MLP, FIR is FIR MLP, IIR is IIR MLP, ACT is activation feedback MLP, OUT is output feedback MLP. Learning rate $\mu = 0.01$.

Fig. 6 Comparison among different architectures performed on the identification of the 16 PAM system. $\mu = 0.01$.

Fig. 7 Comparison among different choices for the truncation parameter ($Q_2 = 0$ is the old instantaneous BPTT) on the same problem of fig. 5. Results for, from top to bottom: IIR MLP, activation feedback MLP, output feedback MLP. $\mu = 0.003$.

Fig. 8 Comparison among different choices for the truncation parameter ($Q_2 = 0$ is the old instantaneous BPTT) on the same problem of fig. 6. Results for, from top to bottom: IIR MLP, activation feedback MLP, output feedback MLP. $\mu = 0.003$.

<table>
<thead>
<tr>
<th>Truncation Parameter ($Q_2$)</th>
<th>Asymptotic Learning MSE [dB]</th>
<th>Testing MSE [dB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (IBPTT)</td>
<td>-18.0</td>
<td>-18.0</td>
</tr>
<tr>
<td>2</td>
<td>-19.0</td>
<td>-19.0</td>
</tr>
<tr>
<td>3</td>
<td>-19.2</td>
<td>-19.1</td>
</tr>
<tr>
<td>4</td>
<td>-20.7</td>
<td>-19.4</td>
</tr>
<tr>
<td>10</td>
<td>-20.8</td>
<td>-19.7</td>
</tr>
<tr>
<td>20</td>
<td>-20.9</td>
<td>-19.6</td>
</tr>
</tbody>
</table>

TABLE 2. Generalization performances on the identification of the 16 PAM system of CBPTT with different values of the truncation term $Q_2$, for a specific IIR-MLP.

TABLE 1. Generalization performances of various locally recurrent neural networks on identifying the 16-PAM transmission system.

<table>
<thead>
<tr>
<th>MLP type</th>
<th>Asymptotic Learning MSE [dB]</th>
<th>Testing MSE [dB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAT</td>
<td>20.04</td>
<td>-20.09</td>
</tr>
<tr>
<td>FIR</td>
<td>-19.27</td>
<td>-19.43</td>
</tr>
<tr>
<td>IIR</td>
<td>23.85</td>
<td>24.39</td>
</tr>
<tr>
<td>ACT</td>
<td>24.39</td>
<td>-24.5</td>
</tr>
<tr>
<td>OUT</td>
<td>23.18</td>
<td>-23.32</td>
</tr>
</tbody>
</table>