ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 379-384

Autonomous learning algorithm for fully connected recurrent networks

Edouard Leclercq, Fabrice Druaux, Dimitri Lefebvre

Groupe de Recherche en Electrotechnique et Automatique du Havre
Université du Havre, Le Havre, FRANCE

Abstract :. In this paper a fully connected RTRL neural network is studied. In order to learn dynamical behaviours of linear-processes or to predict time series, an autonomous learning algorithm has been developed. The originality of this method consists of the gradient based adaptation of the learning rate and time parameter of the neurons using a small perturbations method. Starting from zero initial conditions (neural states, rate of learning, time parameter and matrix of weights) the evolution is completely driven by the dynamic of the learning data. Two examples are proposed, the first one deals with the learning of second order linear process and the second one with the prediction of the chaotic intensity of NH3 laser. This last example illustrates how our network is able to follow high frequencies.

1.Introduction

Recurrent neural networks are very helpful to solve dynamical problems. Properties of dynamical recurrent networks such as oscillatory or chaotic behaviours were studied between 1989 and 1995 [1, 2]. Recently, various applications have been developed using such networks. Numerical series prediction was studied by Chang & al. with application examples such as simulation of a second-order low pass filter and chaotic dynamic of the intensity pulsations of NH_3 laser [3]. This last application was also investigated by M.W.Mak & al. in the context of RTRL (Real Time Recurrent Learning) algorithm improvement [4]. Plant control and identification applications are also often concerned by searchers and engineers works [5, 6, 7, 8].
Various algorithms are used by different authors for adaptation and learning. Among these algorithms let us notice an increasing interest for the RTRL proposed since 1989 by R.J.William and D.Zipser [9, 10]. According to the considered applications, several improvements were brought to this algorithm: the sub-grouping strategy [11], the constrained RTRL and the conjugate Gradient Algorithm [3, 4, 12].
In this paper an original method has been developed for the implementation of the fully connected networks RTRL independently of any initialisation stage. The main advantage of this method is that the dynamic of the system to learn does not have to be exactly known. Starting from zero initial conditions, the proposed network adapts itself, so as the learning rate and the time parameter, in order to track dynamical behaviours. This property is useful for time varying and/or unknown processes. Simulations, for learning of linear processes and numerical series prediction, prove the efficiency of the method.

2.Network structure and dynamics

The proposed network is dynamic, fully connected, continuous and recurrent (figure 1). It is entirely defined by the following equations:

$$\frac{1}{\tau}\frac{dX_i(t)}{dt} + X_i(t) = U_i(t) = \tanh\left(\sum_{j=1}^{N} w_{ij}(t)X_j(t) + I_i(t)\right), \quad (i, j) \in \{1,...,N\}^2 \qquad (1)$$

The function $X_i(t)$ represents the output of the ith node whose external input is $I_i(t)$ and whose activation is $U_i(t)$. The parameter τ is a time-constant. The matrix of weights $W = [w_{ij}(t)]$, $(i, j) \in \{1,\ldots,N\}^2$ is defined for the connections between the neurons. Each entry $w_{ij}(t)$ of the matrix W stands for the weight from the jth neuron to the ith neuron. Each neuron can be an input or an output or both at the same time. All neurons take part of the calculation of the relation between inputs and outputs. Let us define by Out the set of the neurons indices that are considered as outputs and by In the set of the neurons indices which receive external inputs. Let us define the output vector of the network $S(t) = [X_i(t)]^T$, $i \in$ Out, the vector of the desired outputs $S_d(t) = [X_{di}(t)]^T$, $i \in$ Out, and the input vector $I(t) = [I_i(t)]^T$, $i \in$ In.

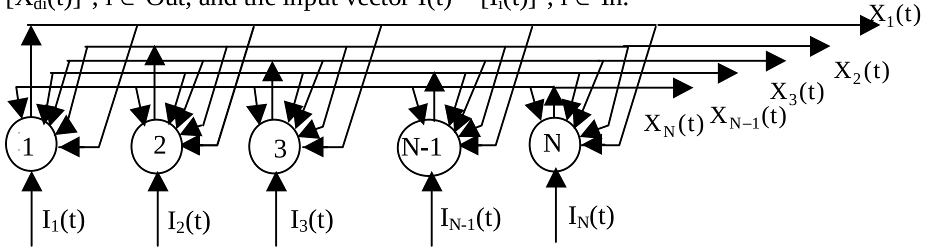


Figure 1. Fully connected MIMO Neural Network.

Let us consider the instantaneous square error between the desired output vector and the network output vector. The RTRL method described by various authors [1, 9, 10, 13] consists in doing the adaptation of the weight matrix W by means of the gradient of the average error. The use of a small perturbations method [12, 13] results in the real time adaptation [10]:

$$\Delta w_{ij}(t) = -\eta\frac{\partial E(t)}{\partial w_{ij}} = -\eta \sum_{k\in Out}(X_k(t) - X_{dk}(t))P_{kij}(t) \qquad (2)$$

$$P_{kij}(t) = \frac{\partial X_k(t)}{\partial w_{ij}}, \quad (k, i, j) \in \{1,..,N\}^3 \qquad (3)$$

From equations (1) to (3), equation (4) is deduced:

$$\frac{1}{\tau}\frac{dP_{kij}(t)}{dt} + P_{kij}(t) = \tanh'\left(\sum_{m=1}^{N} w_{km}(t)X_m(t) + I_k(t)\right)\left(\delta_i^k X_j(t) + \sum_{m=1}^{N} w_{km}(t)P_{mij}(t)\right) \qquad (4)$$

where $\tanh'(Y) = \frac{d\tanh(Y)}{dY}$ and δ_i^k is the Kronecker symbol.

This learning algorithm, as many other ones, requires an initialisation stage [1, 2, 14]. The vector $X(0) = [X_i(0)]^T$, $i \in \{1..N\}$, the weights matrix W and the parameters η and τ are generally randomly selected or deduced from the system dynamics. In the following section, the η and τ initialisation problem is solved through an automatic process starting from zero initial conditions.

3. Automatic adaptation of the parameters τ and η

In order to obtain an automatic adaptation of τ and η, let us define $\tau(t)$ and $\eta(t)$ as:

$$\frac{d\tau(t)}{dt} = -\eta(t)\frac{\partial E(t)}{\partial \tau} \qquad \frac{d\eta(t)}{dt} = -\frac{\partial E(t)}{\partial \eta} \tag{5}$$

These relations have been used in order to adapt these parameters according to the error minimisation such as weigh matrix. Using the relation (2) we can write:

$$\frac{\partial E(t)}{\partial \tau} = \sum_{i \in Out} \left(X_i(t) - X_{di}(t)\right) K_i(t) \quad \text{with} \quad K_i(t) = \frac{\partial X_i(t)}{\partial \tau} \tag{6}$$

Then a small perturbation method increased by a term $\varepsilon/\tau(t)$ allows us to write:

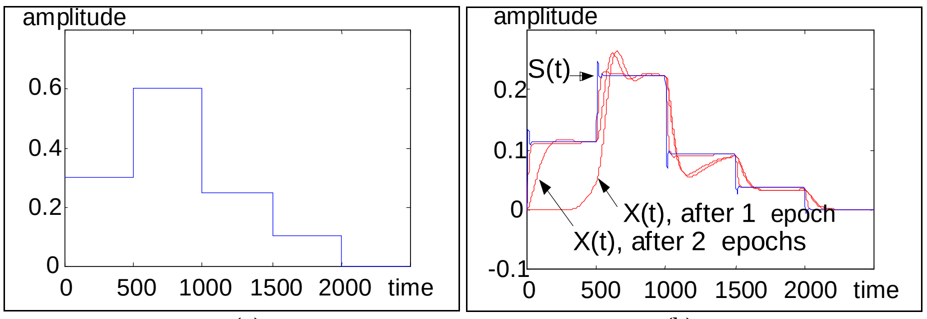
$$\frac{1}{\tau(t)}\frac{dK_k(t)}{dt} + K_k(t) = \tanh'\left(\sum_{m=1}^{N} w_{km}(t)X_m(t) + I_k(t)\right)\left(\sum_{m=1}^{N} w_{km}(t)K_m(t)\right) + \frac{\varepsilon}{\tau(t)} \tag{7}$$

Let us notice that the network evolves in an autonomous way starting from zero initial conditions, according to the value ε, indeed, $dK_k(0)/dt = \varepsilon$. The term $\varepsilon/\tau(t)$ is added in order to avoid that $K_i(t)$ reaches a zero value. The same holds for η. The adaptation of η and τ parameters is obtained from equations (5) to (7). The parameters reach the best values according to the system dynamic.

4.Simulations

In order to test the efficiency of this algorithm, two different problems have been investigated. First, the network learns the dynamic of a SISO second order linear-process. Secondly, the network has been trained to predict the chaotic evolution of the intensity of a NH3 laser time series. Simulations are performed for a various number of delayed inputs that are necessary in order to learn the dynamic of a process, by tacking into account the last n output responses. The total number of neurons depends on the number of delays. The nodes 1 to N-1 are input neurons. The neuron 1 receives the input signal E(t) and the neurons 2 to N-1 receive the delayed output signals S(t-1), S(t-2),…, S(t-N+2). The node N is the output node that provides an estimation of the output signal S(t) at time t. This algorithm acts in such a manner (zero initial conditions) that additional neurons will never evolve from zero.

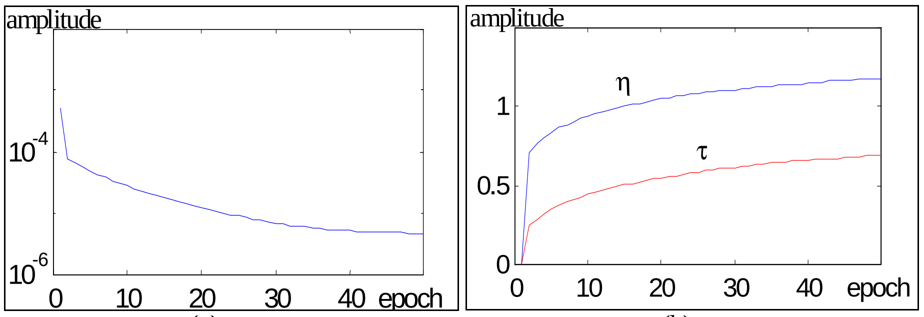
A second order system defined by $H(p) = 3/(p^2+2p+8) = S(p)/E(p)$ is investigated. Using the input signal presented on figure 2, output data are processed for the system. The network is composed of four neurons. The first one receives the input $I_1(t)=E(t)$, the second and third ones are the delayed outputs, respectively $I_2(t)=S(t-1)$, $I_3(t)=S(t-2)$ and the last one is the network output $X_4(t)$.



(a) (b)

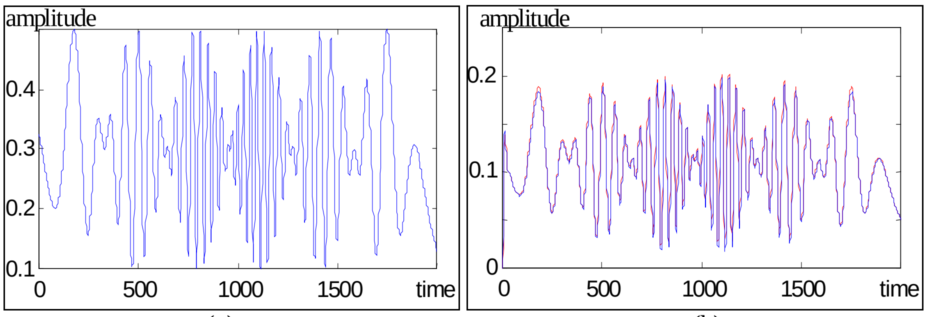
Figure 2. Learning of H(p): (a) Input signal for training. (b) Test of the network response after 1, 2 and 50 epochs of learning.

Data are presented to the network and the weights, η and τ are updated after each presentation, ε is set to 1. The responses of the network carried out for a data set of 2500 samples and the mean squared error (MSE) are presented in figure (2) and (3).



(a) (b)

Figure 3. Learning of H(p) : (a) MSE versus epochs. (b) η and τ versus epochs.



(a) (b)

Figure 4. Validation of the trained neural network: (a) Input signal for the test after 50 learning epochs. (b) Comparison of the responses of the system and network.

The evolution of the MSE is fast during the first few epochs of learning. The final value η=1.2 is higher than the usual one. In [3, 4] this parameter is respectively set to

0.1 and 0.5. The free evolution of η speeds up the learning. So one can assert that a small value of the learning rate is not essential for such a learning task. The growing value of τ proves that an adaptation of the time parameter increases the efficiency of the network as it changes the dynamic of the nodes.
In order to test the quality of the learning, the signal of the figure (4a) has been presented to the network after the 50th epoch of previous learning stage. On the figure (4b) the response of the 2nd order system and the one of the network are compared. For this test the MSE is 0.000024 and the network behaves exactly as the system.

For the time series prediction, a data set of 1000 samples from the Santa Fe Institute Time Series Prediction and Analysis Competition is used [15]. The data concern the evolution of the chaotic intensity pulsations of a far-infrared NH3 laser (figure 5).
In order to compare theses results to those of referenced works [3, 4] a four node network is used to predict the NH3 laser time series. The first node receives the first delayed output $I_1(t) = S(t-1)$, the second and third nodes receive respectively $I_2(t) = S(t-2)$, $I_3(t) = S(t-3)$ and the last neuron is an estimation of $S(t)$. For this simulation ε equals 1.

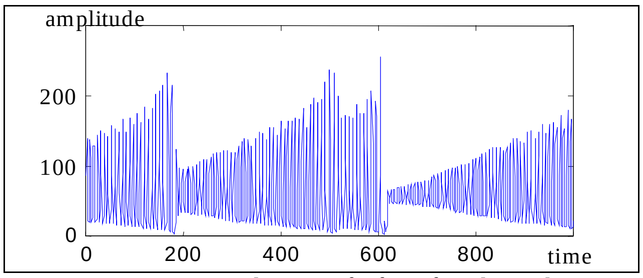


Figure 5. Intensity pulsations of a far-infrared NH3 laser.

The MSE reaches 0.0022 after 80 training epochs. Table 1 presents a comparison of the performances of this algorithm with the results presented in referenced works [3, 4]. For this algorithm, the MSE of 0.0022 is obtained after only 60 epochs of learning while for the others algorithms the presented results were obtained for 1000 epochs (W-F. Chang & al.) and 2000 epochs (M.W.Mak & al.). TE (Time per Epoch) results from the CPU time divided by the number of epochs. The best TE are obtained by the SGRTRL and the presented Autonomous RTRL method (TE=0.0935 and 0.1)

	M.W.Mak & al. [4] P Pro 200MHz		W-F Chang & al [3] P200 MHz		Present algorithm P400 MHz
	RTRL	SGRTRL	RTRL	CGRL	Autonomous RTRL
MSE	0.002046	0.017165	0.00099	0.000565	0.0022
CPU time(s)	456	187	273	1854	8
TE	0.228	0.0935	0.273	1.854	0.1

Table 1 : NH3 laser prediction results : MSE, CPU time and TE.

5. Conclusions

Fully connected RTRL neural networks have been investigated in this paper. The main contribution concerns the dynamic adaptation of the learning rate and time parameter. Such a method has some advantages that have been pointed out for dynamical behaviours learning and time series prediction. The size of the proposed RTRL neural networks depends only on the number of inputs and outputs. Moreover, starting from zero initial conditions, it does not require any initialisation procedure. Finally, comparisons with some related works emphasis the efficiency of the proposed algorithm. Prediction and control design of industrial processes will be considered in our further works. Constraints on weights will also be investigated in order to prevent instability behaviours and over-learning that may occur with some applications. Overfitting problem has to be further studied.

References

1. K. Doya and S. Yoshizawa, "Adaptive neural oscillator using continuous-time back-propagation learning", Neural Networks, 2, 375-385,1989.
2. Y. Hayashi, "Oscillatory neural network and learning of continuous transformed patterns", Neural Networks, 7 (2), 219-231, 1994.
3. W.F. Chang and M.W.Mak, "A conjugate gradient learning algorithm for recurrent neural networks", Neurocomputing 24 (1-3), 173-189, 1999.
4. M.W. Mak, K.W. Ku and Y.L. Lu, "On the improvement of the real time recurrent learning algorithm for recurrent NN", Neurocomputing, 24 (1-3), 13-36, 1999.
5. S. Ellamayar V.T., Y.C. Shin, "State estimation of continuous-time radial basis function networks", Automatica, 36, 399-407, 2000.
6. J.O. Jang, G.J. Jeon, "A parallel neuro-controller for DC motors containing nonlinear friction", Neurocomputing, 30, 233-248, 2000.
7. R-J. Wai, H-H.Lin, F-J.Lin, "Hybrid controller using fuzzy NN for identification and control of induction servo motor drive", Neurocomputing, 35, 91-112, 2000.
8. D. Wang, P. Bao, "Enhancing the estimation of plant Jacobian for adaptive neural inverse control", Neurocomputing, 34, 99-115, 2000.
9. R.J. William and D. Zipser, "Experimental analysis of the recurrent learning algorithm", Connection Science, 1, 87-111,1989.
10. R.J. William and D. Zipser, " A learning algorithm for continually running fully recurrent neural networks, Neural Computation, 1, 270-280, 1989.
11. D. Zipser, "A subgrouping strategy that reduces complexity and seeds up learning in recurrent networks", Neural Computing, 1, 552-560, 1989.
12. F. Druaux, E. Rogue and A. Faure, "Constrained RTRL to reduce learning rate and forgetting phenomenon", Neural Processing Letters, 7, 161-167,1998.
13. K. Doya, "Bifurcations in the learning of recurrent neural networks", Proceedings of the IEEE international symposium on circuits and systems, 2777-2780, 1992.
14. R.J. Williams, "Adaptive state représentation and estimation using recurrent connectionnist networks", in Miller, Satten, Webos, NN for Control, MIT Press, Cambridge, 1990.
15. A.S.Weigend, A. Gershenfeld, "Time series prediction: forecasting the future and understanding the past", eds. Reading, MA: Addison-Wesley, 1994.