1997 IEEE International Symposium on Circuits and Systems, June 9-12, 1997, Hong Kong

Improved signal processing with dynamic recurrent neural models using ARMA-like units

Jean-Philippe Draye°, Davor Pavisic°, Guy Cheron*, Gaëtan Libert°

° Polytechnical Faculty of Mons
"Parallel Information Processing" Laboratory
Rue de Houdain, 9 - B7000 Mons (Belgium)

* Free University of Brussels
Laboratory of Biomechanics
Avenue Heger, 28 - B1050 Brussels (Belgium)

I. Introduction

Artificial neural network research essentially concerns feedforward networks which are now widely considered as powerful tools to approximate functions, to execute classification tasks or to act as an associative memory (these tasks can be considered as *static*). Nevertheless, the last few years, many research studies have concerned recurrent neural networks because of their capabilities to deal with complex temporal tasks and to exhibit rich dynamic behaviour. Some of these recurrent networks can be trained to learn a fixed point behaviour (such as the Hopfield network or the Boltzmann machine). We are concerned with recurrent models that can exhibit a non-autonomous and non-converging dynamics [8]. These latter networks have time-varying inputs and/or outputs and are particularly suitable for adaptative temporal processing such as signal production (motor control), signal recognition (speech recognition), signal prediction (time series prediction) or signal processing (adaptative filtering).

In this paper, we consider a general recurrent neural network model that is fully-connected. Let us point out that our model is governed by continuous-time equations and that we associate to each neuron-like unit an adaptive time constant that is modified by the learning process. The learning algorithm of such a network, called the *Time-Dependent Recurrent Backpropagation (TDRBP)* algorithm, has been derived using the Maximum Principle of Pontryagin of the theory of control [1]. Using this latter principle, we have also elegantly introduced in our network nonlinear autoregressive (AR) and mixed autoregressive-moving average (ARMA) units. The ARMA processes have been introduced by G. Box and G. Jenkins for the forecasting and control of time series [2]. In practice, each inner weights of the network has been replaced by an adaptive AR filter; moreover, a moving average (MA) process has been added to each output units for which a target signal is available. All the parameters of these ARMA processes are adaptive and modified by the TDRBP algorithm.

The work of the first author was supported by the Belgian National Fund for Scientific Research (F.N.R.S.). This work was partly supported by the European Community through the ESPRIT program NeuroCOLT (Working Group Nr 8556). The work of the fourth author was supported by the Research Fund of the University of Brussels.

J.P. Draye is also a Senior Research Assistant of the Belgian National Fund for Scientific Research (F.N.R.S.) and D. Pavisic is with the Universitad Del Valle (Cochabamba - Bolivia).

Theoretical results show that the adaptive time constants improve the long-term capabilities of recurrent neural models and enrich the dynamical behaviour of the network (see [4] for more details).

II. The neural model and its learning algorithm

As we stated in the introduction, we consider a neural model whose individual units are governed by the following continuous-time equations (see [8]) :

$$T_i \frac{dy_i}{dt} = -y_i + F(x_i) + I_i \quad \text{where} \quad x_i = \sum_j w_{ji}\, y_j \tag{1}$$

where y_i is the state or activation level of unit i and T_i is its membrane time constant, w_{ij} the weight associated to the connection out of neuron i to neuron j, $F(\alpha)$ is a squashing function (sigmoid-like function). The term x_i is the total or effective input of the neuron; I_i is an external input (in the following, we will assume that terms I_i will be replaced by adaptative weights w_{0i} connected to a fixed input which is set to one).

As one immediately see, the equations are continuous in time; our model is a *temporal transformation network*. We have thus to differentiate our model to *spatial transformation networks* which identify spatial patterns presented at the input layer and respond with a spatial pattern of activations at the output units (such as the classical task of recognition of handwritten characters). Our network, due to its continuous-time feature, will deal with time-varying signals.

In 1956, a principle, leading to the solution of the general problem of finding a control process was proved by L. S. Pontryagin [1]. This principle gives an interesting framework to solve the general case of minimizing an arbitrary functional of the integral function of variable systems. We thus consider the dynamical system which is described by a system of differentiable equations of the n-th order :

$$\frac{dy_i}{dt} = f_i(y_1, \ldots, y_n \,;\, u_1, \ldots, u_r \,;\, t) \quad i = 1, \ldots, n \tag{2}$$

where $(y_1, \ldots, y_n)$ is the phase point describing the system in the n-dimensional phase space $\mathbb{R}^n$ and $(u_1, \ldots, u_r)$ the control vector.

The optimal problem can be formulated in the following way : ξ_0, ξ_1 are two given points in $\mathbb{R}^n$; from the class of permissible controls, it is necessary to select a control $\mathbf{u}(t), t_0 \le t \le t_1$, for which there is a suitable path $\mathbf{y}(t)$ from equation (I.1), defined over the whole segment $t_0 \le t \le t_1$ and joining the points $\xi_0, \xi_1 : \mathbf{y}(t_0) = \xi_0,\ \mathbf{y}(t_1) = \xi_1$ and

$$Q = \int_{t_0}^{t_1} q(\mathbf{y}, \mathbf{u}, t)dt \tag{3}$$

reduces to a minimum. Depending on the choice of the function $q(\mathbf{y}, \mathbf{u}, t)$, the integral (3) can indicate the expenditure of time, energy, ... during the course of the process under consideration [7]. We define a covariant vector $\mathbf{p}$ of the space $\mathbb{R}^n$ which components p_i (called the adjoint variables) are defined by the partial derivatives of $Q : \mathbf{p} = \nabla_{\mathbf{y}} Q$. We introduce here the famous Hamiltonian function :

$$H = q + \mathbf{p}^T \frac{d\mathbf{y}}{dt} \tag{4}$$

This Hamiltonian is a function of $\mathbf{y}$, $\mathbf{p}$ and $\mathbf{u}$. The solution of the minimization problem is given by the Hamilton-Jacobi equation :

$$\min_{\mathbf{u}} (q + \mathbf{p}^T \frac{d\mathbf{y}}{dt}) + p_0 = 0 \tag{5}$$

This equation is equivalent to $H^* = \min_{\mathbf{u}} H$ which constitutes *the Minimum Principle*. This principle is changed into *the Maximum Principle* by changing the sign of the Hamiltonian (as originally proposed by Pontryagin). Pontryagin has proved that the solution of the Hamilton-Jacobi equation are [1] :

$$\frac{d\mathbf{y}}{dt} = \nabla_{\mathbf{p}} H^* \quad \text{and} \quad \frac{d\mathbf{p}}{dt} = -\nabla_{\mathbf{y}} H^* \tag{6}$$

These latter equations are the Hamilton-Pontryagin equations.

Our goal is to train recurrent neural networks (with n neurons) represented by the following mathematical model (see eq. (1)). Note that, for this first step, we do not take the time constants into account.

We will use the Maximum Principle to demonstrate the learning equations and, we have thus to introduce some new variables p_i called the adjoint variables. The goal is to find the vector command (in our case the weights matrix) that minimizes the cost function E. With the previous definitions, we see that the Hamiltonian (4) can be developed in :

$$H = q + p_1\ [-y_1 + F(x_1)] + \ \dots\ + p_n\ [-y_n + F(x_n)] \tag{7}$$

Using the Pontryagin Maximum Principle, we can derive the adjoint variables :

$$\frac{dp_i}{dt} = p_i - e_i - \sum_j w_{ij}\ F'(x_j)\ p_j \tag{8}$$

where $e_i(t) = \frac{\partial q(\mathbf{y}(t), t)}{\partial y_i(t)}$. The following learning equation that gives the corrections to apply to the network weights :

$$\frac{\partial E}{\partial w_{ij}} = \int_{t_0}^{t_1} y_i\ F'(x_j)\ p_j\ dt \tag{9}$$

If we introduce the adaptative time constant T_i associated to each neuron, the mathematical model becomes : $T_i \frac{dy_i}{dt} = -y_i + F(x_i)$. Using the Maximum Principle, we can find the adaptation equation for the time constants :

$$\frac{\partial E}{\partial T_i} = -\frac{1}{T_i} \int_{t_0}^{t_1} p_i\ \frac{dy_i}{dt}\ dt \tag{10}$$

III. Neural model with ARMA-like units

In this section, we will introduce some modifications to our network by modeling each link between two units by an autoregressive (AR) filter. The original aspect of our approach is the fact that our model is governed by continuous equations and can handle dynamic tasks. Moreover, as we will see, the approach based on the Pontryagin Maximum Principle offers a general framework for the derivation of the learning algorithms of very general continuous-time neural networks.

We introduce new connection weights denoted w_{ij}^k (i.e. the autoregressive weight between unit i and j from the $k-th$ delay link. By coherency with the notation of Box and Jenkins [2], we will note p the maximum connection delay (which can be considered as the order of the autoregressive filter) (see Figure 1a). Accordingly, if a target signal is available for a particular unit (as it is the case for the output units in a supervised learning process, this signal is denoted $t_j(t)$ on Figure 1b), we add a moving-average process of order q (see Figure 1b). The new connection weights are denoted v_j^k (i.e. the moving-average weight of unit j from the k-th delay link).

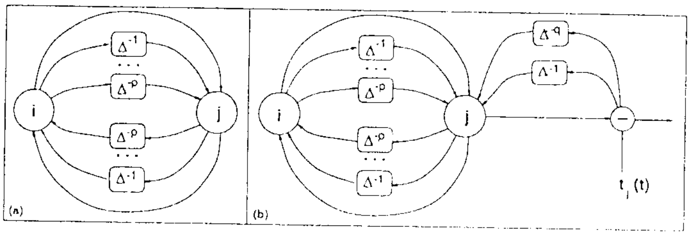

Fig. 1. *Illustration of the connections between two neurons for an AR process (a) and an ARMA model (b).*

The new learning equations for the adaptation of parameters w_{ij}^k and v_j^k are given by deriving the derivatives of the new Hamiltonian function. Using equations (6) and (7), we can derive the adjoint variables :

$$\frac{dp_i}{dt} = p_i - e_i - \sum_{j=1}^{N}\sum_{k=0}^{p} p_j(t + k\Delta t)F'(x_j(t + k\Delta t))w_{ij}^k - \sum_{k=1}^{q} p_i(t + k\Delta t)F'(x_i(t + k\Delta t))v_i^k \tag{11}$$

and the corresponding learning equations :

$$\frac{\partial E}{\partial w_{ij}^k} = \int_{t_0+k\Delta t}^{t_1} y_i(t-k\Delta t)\ F'(x_j(t))\ p_j\ dt \quad (12)$$

$$\frac{\partial E}{\partial v_i^k} = \int_{t_0+k\Delta t}^{t_1} [y_i(t-k\Delta t) - t_i(t-k\Delta t)]\ F'(x_j(t))p_j(t)\ dt \quad (13)$$

This extended version of the algorithm of *Time-Dependent Recurrent Backpropagation* allows to adapt all the different parameters of the model. It is also based on continuous-time equations. Moreover, one has to note the interest of the Pontryagin Maximum Principle to derive the learning algorithms of complex neural architectures.

IV. Determination of the architecture of the network

It is obvious that the introduction of new parameters will increase the complexity the training process and increase the learning time. In order to find a good compromise between the quality of the learning and its duration, it is important to choose carefully the number of neurons in the network *as well as* the orders of the autoregressive filters (i.e. p) and of the moving-average processes (i.e. q).

As it is generally accepted, the choice of the number of the network's neurons is more an art than a science. Some simulations quickly give an idea of the precise number of neurons needed.

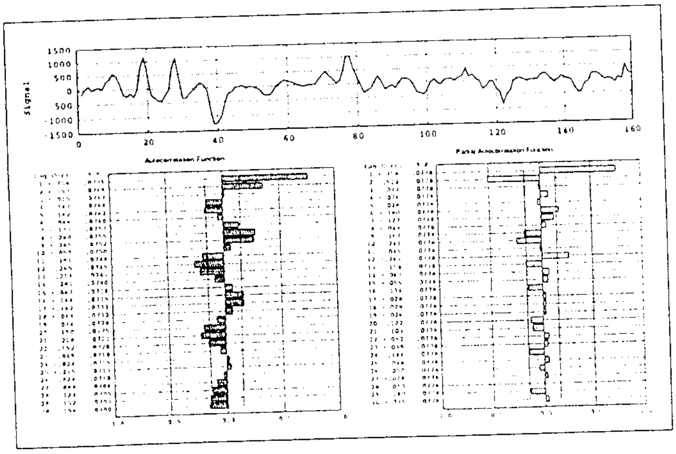

Fig. 2. *Example of continuous time signal to be treated (top) and its respective autocorrelogram (bottom, left) and partial autocorrelogram (bottom, right).*

On the contrary, as far as the ARMA parameters are concerned, several efficient techniques have been developed to estimate the number of parameters required by the processes (see [6]). The major tools used in the identification of the ARMA model are plots of the training signals, correlograms of autocorrelation and partial autocorrelation. The decision is not straightforward and in less typical cases requires not only experience but also a good deal of experimentation with alternative models. However, an ARMA process can be determined among five basic models that can be identified based on the shape of the autocorrelogram (ACF) and partial autocorrelogram (PACF) (for example, an one-autoregressive parameter model is chosen when the ACF exhibits an exponential decay and the PACF has a spike at lag 1 and there is no correlation for other lags, for more details, see [2], [6]).

For example, to treat the continuous-time signal depicted on Figure 2(top), we see that its ACF presents a sine-wave shape pattern and its PACF exhibits two spikes at lags 1 and 2; the corresponding model is a pure autoregressive process with AR parameters (i.e. $p = 2$ and $q = 0$).

V. Simulation results

To illustrate the interest, we will give an example that we have developped in the field of biomechanics. We consider the application of the neural networks to the identification of the kinematics of the human arm during complex movements in free space. A conventional identification system to handle this task is very difficult to design. Indeed, this one would have to take all the concepts of biological motor control into account but even like that, the quality of the identification would be poor because we still ignore many of these concepts (such as the real pathway of informational signals between the muscles and the central nervous system or some movement invariances). Several techniques have been proposed to solve this complex problem using techniques such as the theory of optimization, the identification using mathematical high-order functions or statistical correlation between muscles command signals (i.e. the electromyographic signals or EMG) and limb movements. Unfortunately, all these techniques require important approximations on the EMG signals and/or provide poor simulation results.

For our experiments, four male right-handed subjects between 21 and 25 years (mean weight : 73 kg and mean height : 179 cm) were asked to draw as fast as possible four series of figures eight with the right extended arm in free-space (the initial directions of the movements were up-right, up-left, down-left, down-right respectively). They were asked to perform the movement repetitively (5-10 cycles) at a self-determined frequency (generally from 0.7 to 1.2 kHz). The movements of the arm were recorded and analyzed using the optoelectronic *ELITE* system (including 2 TV cameras working at a sampling rate of 100 Hz [5]). Surface EMG patterns of seven muscles were measured using telemetry. The network that we consider includes 20 neurons; among the twenty fully connected neurons of the DRNN, eighteen receive the inputs (all the different EMG signals) and two of them give the output (the coordinates Y and Z, see Figure 3).

We have proved elsewhere that this model is successful in identifying the complex mapping between full-wave rectified EMG signals and upper-limb trajectory. We have shown that the quality of the identification of the mapping allows to clearly interpret the role of each muscle in any particular movement (more details can be found in [3]).

The important issue for the present paper is to examine the effect of the introduction of ARMA-like units in the

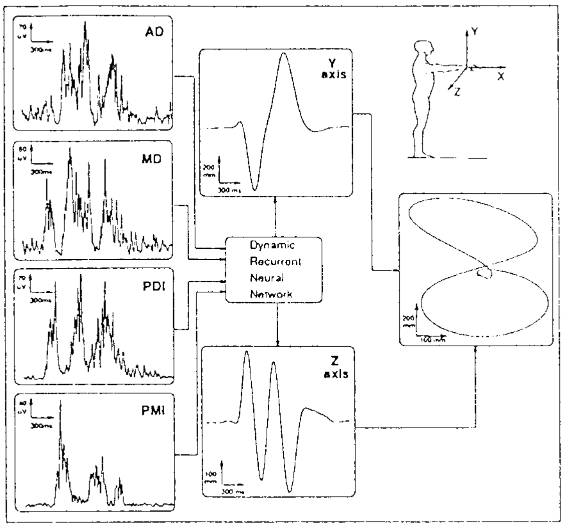

Fig. 3. *Input-output organization of the DRNN. The inputs consist of seven full-wave rectified EMG signals (four of them are depicted). The outputs are the Y and Z coordinates of the index marker during the drawing of the figure eight (shown on the right) with the extended arm.*

network on the quality and the speed of the identification.

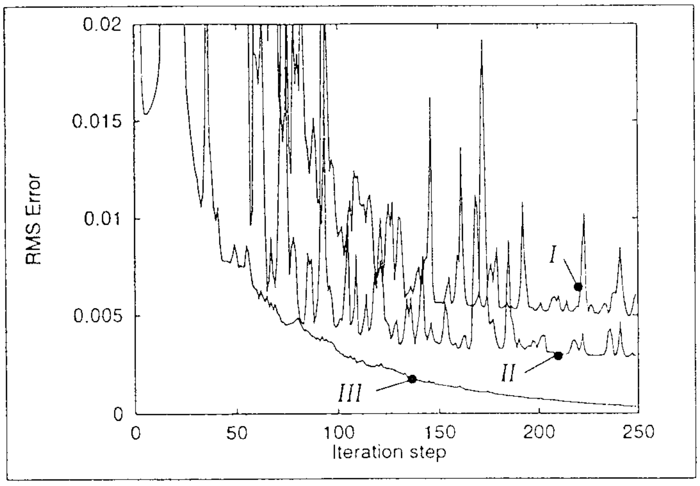

Fig. 4. *Error plots for the training of three different architectures of neural networks : I. 20-neuron network with ARMA processes, II. 20-neuron network with AR connections (order $p = 5$) III. 20-neuron network with ARMA processes (orders $p = 5$ and $q = 2$). Each curve is obtained by averaging 50 different learning curves for every architecture.*

Figure 4 presents the error for the training of three different architectures of neural networks : *I.* 20-neuron network with ARMA processes, *II.* 20-neuron network with AR connections (order $p = 5$) *III.* 20-neuron network with ARMA processes (orders $p = 5$ and $q = 2$). Each error plot has been obtained by averaging the error curves of 50 different learning phases for each type of architecture. We see that the introduction of autoregressive filters increase the speed and the quality of the learning but we also note that the bifurcations are still present. Accordingly, the ARMA processes also decrease the the learning time and hugely improve the quality of the identification but they also suppress the bifurcations that are present in the two other learning curves.

We conclude that the networks obtained by modeling each synapse by an autoregressive or mixed autoregressive-moving average filters can more efficiently model dynamical systems. These extra-connections increase the internal memory capacity of the network to store and update context information.

VI. Conclusion

We have shown that dynamic recurrent neural networks with ARMA-like units can tackle the problem of complex signal processing. In some cases of very highly nonlinear processing, their use can even be inevitable.

We have shown that the Pontryagin Maximum Principle (from the theory of control) helps to elegantly derive the continuous-time learning algorithms for these complex neural architectures.

Finally, we have presented a practical biomedical application where dynamic recurrent networks exhibit their robustness. We are currently investigating other applications in the field of mathematics (such as interpolation tasks i.e., for the forecasting of stock market value) and of engineering (active noise control).

References

[1] Boltyanskii, V.G., Gamkrelidze, R.V., Mishchenko, E.F., and Pontryagin, L.S. The Maximum Principle in the theory of optimal processes of control. In *Optimal and Self-organizing Control*, pages 262–266. R. Oldenburger (Editor), 1966.

[2] Box, G.E.P. and Jenkins, G.M. *Time series analysis : forecasting and control.* Holden-Day Inc., San Francisco, 1976.

[3] Cheron, G., Draye, J.P., Bourgeois, M., and Libert, G. Dynamical neural network identification of electromyography and arm trajectory relationship during complex movements. *IEEE Transactions on Biomedical Engineering*, 43(5):558–552, 1996.

[4] Draye, J.P., Pavisic, D., Cheron, G., and Libert, G. Dynamic recurrent neural networks : a dynamical analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, 26(5), 1996.

[5] Ferrigno, G. and Pedotti, A. ELITE : a digital dedicated hardware system for movement analysis via real-time TV signal processing. *IEEE Transactions on Biomedical Engineering*, 32(11):943–950, 1985.

[6] Libert, G. An automatic procedure for Box-Jenkins model building. *Europena Journal of Operational Research*, 17:95–103, 1984.

[7] Naslin, P. *Théorie de la commande et conduite optimale.* Bibliothèque de l'Automaticien, Dunod, Paris, 1969.

[8] Pearlmutter, B.A. Learning state space trajectories in recurrent neural networks. *Neural Computation*, 1:263–269, 1989.