ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 31-36

Adaptative Time Constants Improve the Dynamic Features of Recurrent Neural Networks

Jean-Philippe DRAYE*,1 , Davor PAVISIC*,
Guy CHERON†, Gaëtan LIBERT*

* Faculté Polytechnique de Mons - *P.I.P.* Laboratory (Belgium)

† University of Brussels - Laboratory of Biomechanics (Belgium)

Abstract. We consider in this paper the improved features of recurrent neural networks where we associate to each neuron-like unit an adaptative time constant T_i. In order to quantify the effects of the T_i's on the network, we present new results using a latching test (to evaluate the long-term memory capabilities) and a study of the modification of the stability regions with the time constants. Finally, a practical application of the Mackey-Glass chaotic signal prediction is presented.

1. Introduction

The last few years, many researchers have been attracted by recurrent neural networks because of their amazing capability to exhibit complex dynamic behaviour. They can be trained to exhibit a fixed point behaviour (the network evolves toward a fixed state such as Hopfield network or Boltzmann machines) or to exhibit a non-autonomous non-converging dynamics (the networks behave as oscillators or as finite automata). Neural networks of the second category 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 will present new and innovative results to quantify the effect of adaptative time constants on the capabilities of recurrent neural models.

2. Recurrent neural model and learning algorithms

We consider a general recurrent neural network model governed by the following equations :

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

where y_i is the state or activation level of unit i, $F(\alpha)$ is a squashing sigmoid-like, I_i is an external input (or bias) and x_i is called the total or effective input of the neuron.

[1]Research Assistant of the Belgian National Fund for Scientific Research.

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 31-36

Let us point out that our model is governed by continuous-time equations and that we associate to each neuron-like unit an adaptative time constant that takes part in the learning process. Equations (1) define the most general recurrent model (except the fact that we do not consider time-delay connections). More simple models can be derived by discretization. For example, if we use a time-step Δt equals to 1 with all the time constants T_i equals to 1, the discretization of equations (1) gives the simple recurrent model proposed by Williams and Zipser [6].

To train a network governed by equations (1), one can use either the *Real-Time Recurrent Learning (RTRL)* algorithm presented by Williams and Zipser [6] or the *Time-Dependent Recurrent Backpropagation (TDRBP)* algorithm derived by Pearlmutter [5]. In the *TDRBP* algorithm, the error gradients are computed from the differential equations (and are thus continuous in time). The resulting equations then must be numerically integrated for simulation. In the *RTRL* algorithm, a different approach is adopted by first discretizing the differential propagation equation (1) and then calculating the gradients from the discretized equations. *RTRL* and its variants have been called *forward gradient* methods, as opposed to *TDRBP* which is a *backward gradient* method.

3. Theoretical results

This section will present new theoretical results concerning the association of an adaptative time constant to each neuron-like unit of a recurrent neural network.

The general task of temporal processing held by recurrent neural networks brings out the necessity for the system to effectively deal with the temporal nature of the incoming data, that is to allow a past input to properly exert its effect at a subsequent time. Generally, a distinction is made between two types of memory mechanisms according to the time interval : *short-term* or *long-term* memory. In the field of neural systems, we can distinguish two types of behaviours that are the counterpart of the types of memory mechanisms. First, we have those systems in which the effect of their past inputs decays to zero during time; they exhibit a *forgetting behaviour* (in absence of external input, their evolution asymptotically converges to a unique point, typically the origin, thus nullifying any previously occurred computation). These models are successfully employed for short-term memory requirements. On the other side, some models are capable of maintaining the effect of their past inputs over an arbitrary long interval of time. Such models, including dynamic recurrent neural networks, are said to exhibit a *latching behaviour.*

To handle temporal processing, a dynamical system must obviously exhibit a latching behaviour but, moreover must satisfy three requirements to be really efficient : *(i)* it must be able to store information for an arbitrary duration, *(ii)* it must be resistant to input noise, and *(iii)* its parameters must be learnable in reasonable time.

3.1 The latching test

The latching test, proposed by Frasconi et al [3], has been designed to test if the

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 31-36

three conditions enumerated above are satisfied.

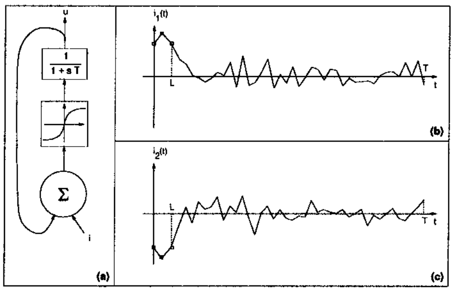

Figure 1: Latching test. a shows the latching single recurrent neuron. b and c show the two trainable inputs.

For this task, the dynamic system under study has to learn discriminating between two different sets of sequences, whose class should be determined by the values of the input on a fixed number L of time steps at the beginning of each sequence of length T (see Figure 1). If we allow sequences of arbitrary length, then the problem can be solved only if the dynamic system is able to latch information about the initial input values.

The initial input $i(t)$ for $t \leq L$ are learnable parameters whereas $i(t)$ is zero-mean Gaussian noise for $t > L$. Optimization is based on the mean square error with targets close to 1.0 and -1.0 for the last time step T.

A set of simulations were carried out to evaluate the effectiveness of dynamic recurrent networks on this simple task.

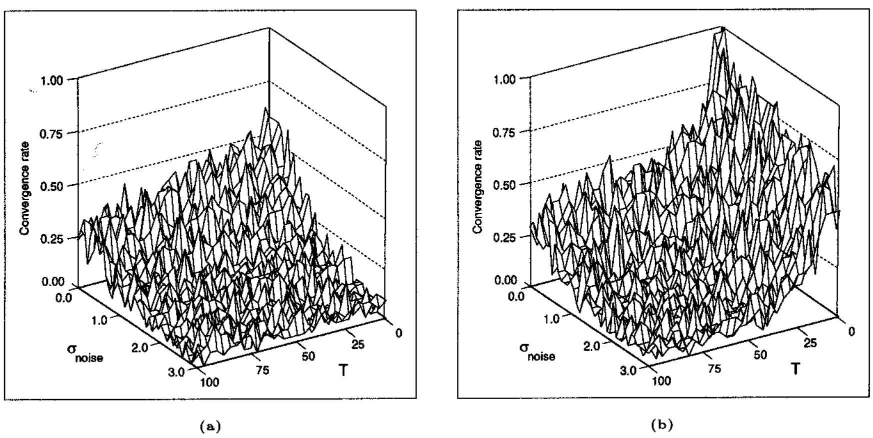

Figure 2: Convergence rate of the learning for a single-recurrent network in the case of the latching test. The convergence rate is plotted versus the standard deviation σ_{noise} of the noise and the length of the sequence T. (a) single-neuron network without time constant, (b) with time constant.

We investigated the effect of the noise variance σ_{noise} and of the sequence length T. A density plot of convergence is depicted on Figure 2, each value was averaged over 30 runs for each of the selected pairs (σ_{noise}, T) (L was chosen equal to 3). Figure 2a is associated to a simple recurrent neuron without adaptative time constant, whereas Figure 2b is associated to the same neuron with an adaptative time constant.

It is clear from Figure 2 that long-term dependencies are much efficiently learned

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 31-36

with a dynamic neural network. Moreover, the addition of adaptative time constants significantly improves the robustness of the network in front of an increase of the standard deviation σ_{noise} of the noise.

3.2 Stability regions

Another original way to characterize the influence of T_i's on the network dynamics is to compute the stability regions corresponding to some particular equilibrium points of a learned trajectory and to study their evolutions with respect to time constants. Indeed, the success of regenerating a desired temporal behaviour, leading to a particular final state, from partial information is directly related to the stability boundaries of attraction of the corresponding final state. This success, as we will see, depends on the time constant values.

We are interested here in the stability region of a locally stable equilibrium point $\mathbf{y}^*$ of a dynamic recurrent neural network. We will compute these regions using the classical Lyapunov functions [2].

In order to get exploitable results, we applied this method to a network of three neurons; the network was limited to such a small size just to allow the visualization of the stability regions in the phase-space $\mathbb{R}^3$ of the activations (y_1, y_2, y_3). We trained the network to follow a trajectory starting at an initial state of the three neurons and to evolve alone to the desired fixedpoint corresponding to the initial state.

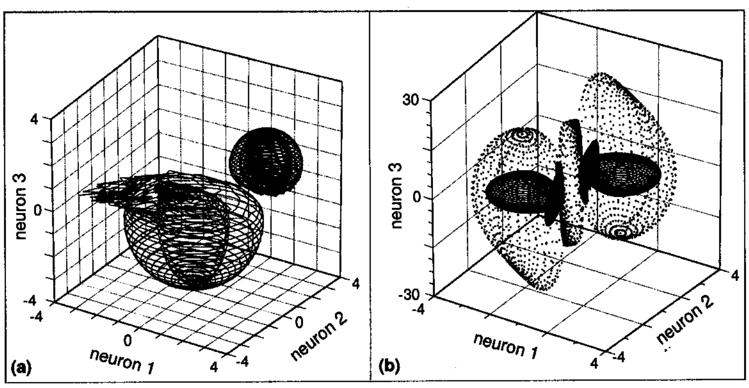

Figure 3: Stability regions computed with Lyapunov functions. (a) for a two fixedpoints system; (b) for a three fixedpoints system.

Figure 3a presents the stability regions associated to a two-fixedpoint system (base of the training set of the three-neuron network). After the training, we modified the time constants in a decreasing way to examine the effect of the time constant values on the network dynamics (being careful that the network still exhibits the same fixedpoints behaviour). Figure 3a shows the stability regions associated to both fixedpoints and presents the comparison between the stability regions of the system with normal and small time constants. Inner regions are always associated with small T_i's. The axis labels are the respective activations

y_i of the three neurons. It is clear that the first effect of small time constants is to make the stability regions shrink .

Figure 3b presents the stability regions associated to a three-fixedpoint system (dotted regions). Once again, after the training period, the time constants are decreased; the first effect is the same : the stability regions also shrank. Nevertheless, if we go on decreasing the T_i's, we remark that the central fixedpoint disappears and gives birth to two distinct different fixedpoints.

We see that a simple variation of the T_i's (without any modification of the weights) can modify the temporal tasks learned by a dynamic recurrent neural networks. The time constants can enrich the behaviour by adding and/or modifying the stability regions of the fixedpoints. Such conclusions can be held for the limit cycles or the strange attractors that the network exhibit.

4. Results

We will now present an practical application in order to investigate the impact of adaptative time constants on the performance of recurrent neural networks during chaotic signal prediction. Signal prediction is the classical task where the input to the network is the time-varying signal and the desired output is a prediction of the signal at a fixed time increment in the future.

The test signal that we consider is the famous chaotic signal produced by integrating the Mackey-Glass delay-differential equation [4] :

$$\frac{dx(t)}{dt} = -b\, x(t) + a\, \frac{x(t-\tau)}{1+[x(t-\tau)]^{10}} \tag{2}$$

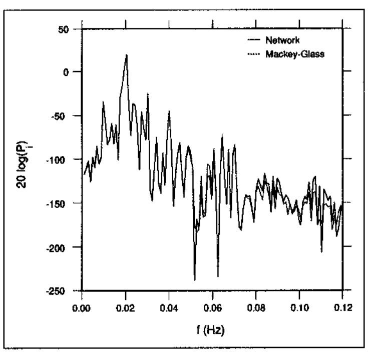

Figure 4: Comparison between the spectrum of the Mackey-Glass signal and the spectrum of the predicted signal produced by the network.

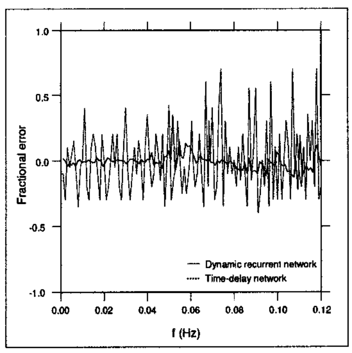

Figure 5: Fractional error between the real and the generated spectra. Solid line is associated to the DRNN, dotted line to a time-delay neural networks

This signal provides a useful benchmark for testing predictive techniques. For comparison with previous techniques, we chose $\tau = 17$, $a = 0.2$, $b = 0.1$, and trained the network to predict six time units into the future.

In order to evaluate the network performance, Figure 4 shows a comparison between the power spectrum of the signal generated by the network and the spectrum of the Mackey-Glass signal[2]. Figure 4 presents the fractional error between the real spectrum and the one produced by a network trained with continuous-time temporal backpropagation with adaptable time delays [1]. The fractional error is defined as the difference between the two spectra, divided by the magnitude of the Mackey-Glass spectrum. This latter network presents a structure of feedforward network with two hidden layers of 10 neurons, one output and a total of 150 adaptable connections. We clearly see that our dynamic recurrent neural network outperform the time-delay neural model : the RMS value of the fractional error falls from 0.252 to 0.064.

5. Conclusion

In this paper, we have shown the interesting new capabilities of recurrent neural network where an adaptative time constant is associated to each neuron. The latching test has quantified the improvements on the long-term memory of the model. Moreover, the computation of the stability regions have shown the enrichment of the dynamical behaviour due to the T_i's. Several applications were successfully developped using dynamic recurrent neural networks including time series prediction, systems identification and control, biomedical applications.

Acknowledgements

This work is supported by the Belgian National Fund for Scientific Research, by the Research Fund of the University of Brussels (U.L.B.) and by the EC through the ESPRIT program NeuroCOLT (Working Group Nr 8556).

References

[1] S.P. Day and M.R. Davenport. Continuous-time temporal backpropagation with adaptable time delays. Technical report, University of British Columbia, 1991.

[2] J.P. Draye, D. Pavisic, G. Cheron, and G. Libert. Dynamic recurrent neural networks : a dynamical analysis. *IEEE Transactions on Systems, Man, and Cybernetics.* in press.

[3] P. Frasconi, M. Gori, M. Maggini, and G. Soda. Unified integration of explicit rules and learning by example in recurrent networks. *IEEE Trans. on Knowledge and Data*, 7(2):340–346, 1995.

[4] M.C. Mackey and L. Glass. Oscillations and chaos in physiological control systems. *Science*, 197:287–289, 1977.

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

[6] R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1:270–280, 1989.

[2] We integrated (2) using a four-point Runge-Kutta method with a step of 0.05. The initial conditions were $x(t) = 0.8$ for $t < 0$, and the equation was integrated up to $t = 1{,}000$ to allow transients to die out.The resulting signal is quasi-periodic with a characteristic time of $T_c = 50$, lying on a strange attractor with a fractal dimension of approximately 2.1.