

## ARTIFICIAL NEURAL NETWORK LEARNING OF NONSTATIONARY BEHAVIOR IN TIME SERIES

MARÍA I. SZÉLIGA, PABLO F. VERDES

PABLO M. GRANITTO and H. ALEJANDRO CECCATTO

*Instituto de Física Rosario, Consejo Nacional de Investigaciones Científicas y Técnicas and  
Universidad Nacional de Rosario, Boulevard 27 de Febrero 210 Bis,*

*2000 Rosario, Argentina*

*szeliga@ifir.edu.ar*

*granitto@ifir.edu.ar*

*verdes@ifir.edu.ar*

*ceccatto@ifir.edu.ar*

We refine and complement a previously-proposed artificial neural network method for learning hidden signals forcing nonstationary behavior in time series. The method adds an extra input unit to the network and feeds it with the proposed profile for the unknown perturbing signal. The correct time evolution of this new input parameter is learned simultaneously with the intrinsic stationary dynamics underlying the series, which is accomplished by minimizing a suitably-defined error function for the training process. We incorporate here the use of validation data, held out from the training set, to accurately determine the optimal value of a hyperparameter required by the method. Furthermore, we evaluate this algorithm in a controlled situation and show that it outperforms other existing methods in the literature. Finally, we discuss a preliminary application to the real-world sunspot time series and link the obtained hidden perturbing signal to the secular evolution of the solar magnetic field.

### 1. Introduction

Nonlinear methods for time series analysis are generally based on the assumption of stationarity. That is, the record has to be originated by an autonomous, non-changing system, so that, for all  $n$ , the joint probabilities  $P(x_{t_1+t}, x_{t_2+t}, \dots, x_{t_n+t})$  of the observation variable  $x$  at arbitrary times  $t_1, \dots, t_n$  are independent of  $t$ . In practice, however, most real-world time series have some degree of nonstationarity due to external perturbations and/or changes in the internal parameters of the system. On the other hand, even if this condition is formally met, natural dynamics are often complex enough to comprise multiple time scales, so that for the naturally limited time span of observations the effective degrees of freedom with the largest scales (of the order of the record length) act as external perturbations for the fastest observed modes.

As a consequence of this, recent works have considered the extension of nonlinear methods to cope

with different problems posed by nonstationary time series.<sup>1,2,3</sup> Some of these works have addressed the proper characterization of nonstationarity,<sup>4</sup> the detection of abrupt discrete changes in the dynamics,<sup>5</sup> the extension of delay embedding ideas,<sup>6</sup> etc. From a practical point of view, applications range from monitoring physiological and mechanical signals<sup>7</sup> to extracting messages from a chaotic background.<sup>8</sup>

In particular, a problem that has received much attention is the reconstruction of slow external driving forces or drifting parameters responsible for the nonstationary behavior in time series. This problem has been discussed by different authors<sup>2,3,9</sup> using methods from the theory of dynamical systems. In a previous work,<sup>10</sup> we have introduced an artificial neural network (ANN)-based method to simultaneously learn the evolution of a slow perturbation responsible for the nonstationary behavior and the intrinsic (stationary) dynamics. Here we refine and complement these ideas, and assess the performance

of the proposed algorithm by comparing it with the results of Ref. 9. We also briefly discuss a preliminary application of our method to the real-world sunspot time series, which allows us to uncover secular changes in solar dynamics.

## 2. ANN Learning of Nonstationary Signals

Lets consider a general regression problem in which an observational nonstationary record  $D = \{x_t, t = 1, N + 1\}$  is modelled by

$$x_{t+1} = f(\mathbf{x}_t, \alpha_t) + \varepsilon_t. \quad (1)$$

where  $\mathbf{x}_t = (x_t, x_{t-1}, \dots, x_{t-d+1})$ , time  $t$  is measured in units of the lag  $\tau$  between observations, and  $\varepsilon_t$  is some residual additive noise of zero mean. The parameter  $\alpha_t$  accounts for the nonstationary changes, caused by either a slow external perturbation acting on the system or internal parameters or degrees of freedom varying on large time scales  $T$  not modelled by  $f$ .

As stated in the Introduction, we will use ANNs to reconstruct both the intrinsic dynamical function  $f$  and the nonstationary signal  $\alpha$  from the available data  $D$ .<sup>10</sup> This program can be carried out as long as  $\alpha$  varies on time scales  $T \gg \tau$ , which is the basic assumption of our method. Furthermore, as a little reflection on this problem indicates, the reconstruction of  $\alpha$  is possible up to a linear transformation, i.e., only the  $\alpha$  profile can be obtained. With these limitations in mind, the intuitive ideas behind our method are the following: Standard modelling of a series requires training a ANN like the one shown in Fig. 1, including only the units represented by full-line circles. This can be accomplished by minimizing with respect to weights and biases  $\mathbf{w}$  the normalized error

$$E_{\text{mod}} = \frac{1}{N\sigma_x^2} \sum_{t=1}^N (x_{t+1} - o(\mathbf{x}_t, \alpha_t = 0; \mathbf{w}))^2. \quad (2)$$

Here  $o(\bullet; \mathbf{w})$  is the network output and  $\sigma_x^2$  the data variance. In this case the input  $\alpha_t$  corresponding to the extra neuron in the input layer (represented by the dashed-line circle in Fig. 1) is set to 0. Although some validation may be required to avoid overfitting, for simplicity we will assume here that the ANN architecture has been chosen not flexible enough to account for noise, so that the training process can be carried out until convergence.

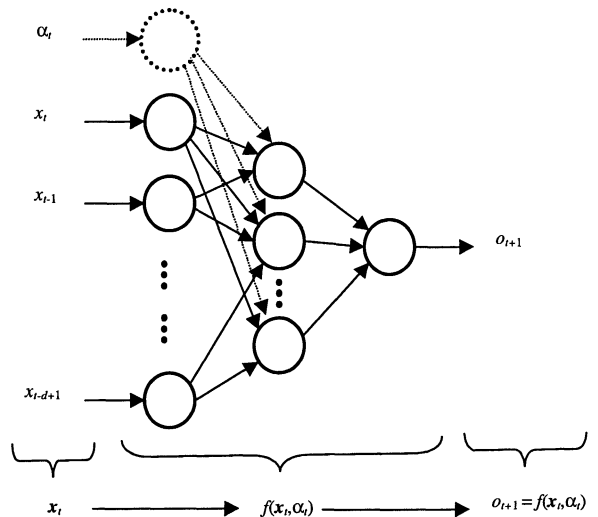


Fig. 1. Feedforward artificial neural network with the extra input unit used to model the  $\alpha$  parameter change.

If we knew the external signal, it would be natural to model  $f(\mathbf{x}_t, \alpha_t)$  by activating the extra input unit in the ANN and feeding it with  $\alpha_t$ . However, since this signal is not known *a priori*, it seems reasonable to adjust its values along the training process by the steepest-descent algorithm, very much like what we do to find weight and biases. The difference with the standard backpropagation algorithm will be the fact that, in this case, one has to minimize with respect to an input to the network. Moreover, since there is a different value  $\alpha_t$  for each training pattern  $(\mathbf{x}_t, x_{t+1})$ , one can imagine that the minimization of the standard quadratic error  $E_{\text{mod}}$  will select these values as to exactly compensate the target-output differences (thus driving  $E_{\text{mod}}$  to zero), instead of reconstructing the correct driving signal profile. This would produce a rapidly changing  $\alpha_t$  perturbation, correlated to the residual errors obtained by training the ANN without this extra input. However, such a result contradicts the basic assumption regarding the smooth behavior of  $\alpha_t$  on time scales of the order of  $\tau$ , the intrinsic dynamics scale. Then, to avoid this possibility one must add to the standard quadratic error a smoothing term for  $\alpha_t$ , like, for instance,

$$E_{\text{smooth}} = \frac{1}{N\sigma_x^2} \sum_{t=1}^N (\alpha_t - \alpha_{t-1})^2. \quad (3)$$

This smoothing error term will penalize sudden  $\alpha_t$  variations (Notice that we have normalized it using

the data variance  $\sigma_x^2$ ). Thus, the total error to be minimized becomes  $E_{\text{tot}} = E_{\text{mod}} + \lambda E_{\text{smooth}}$ . The parameter  $\lambda$  sets the appropriate scale between  $E_{\text{mod}}$  and  $E_{\text{smooth}}$ , and its proper value remains to be determined. If  $\lambda$  is too large, the minimization of  $E_{\text{tot}}$  will first reduce the term  $\lambda E_{\text{smooth}}$ , producing a constant  $\alpha_t$ . Then, it will adjust  $\mathbf{w}$  to minimize  $E_{\text{mod}}$ , obtaining the same global modelling as in the standard approach. On the contrary, if  $\lambda$  is too small the term  $\lambda E_{\text{smooth}}$  becomes irrelevant, and the minimization process will adjust both  $\mathbf{w}$  and  $\alpha_t$  to reduce  $E_{\text{mod}}$ , choosing the external signal as to compensate the uncorrelated errors between  $x_{t+1}$  and  $o(x_t, \alpha_t = 0; \mathbf{w})$ . In Ref. 10 we showed that between these two limits there is a range of  $\lambda$  values for which  $E_{\text{smooth}}$  becomes weakly sensitive to this parameter. Within this range all the  $\alpha$  profiles are similar, and should correspond to the hidden perturbing signal we are seeking to reconstruct. According to the intuitive ideas described above, we propose the following algorithm to determine  $\alpha_t$ :

1. Train to convergence the ANN shown in Fig. 1, keeping  $\alpha_t = 0$  while minimizing with respect to weights and biases  $\mathbf{w}$  the normalized error  $E_{\text{mod}}$ .
2. Switch on the nonstationary perturbation  $\alpha_t$ , and retrain the network minimizing the error  $E_{\text{tot}} = E_{\text{mod}} + \lambda E_{\text{smooth}}$  with respect to  $\mathbf{w}$  and the unknown  $\alpha_t$ 's.
3. Repeat the previous step for different values of the scale parameter  $\lambda$ , and plot the obtained errors  $E_{\text{smooth}}$  at the minimum of  $E_{\text{mod}}$  as a function of  $\lambda$ . A consideration of this curve should indicate the adequate value of  $\lambda$  and, consequently, the correct  $\alpha$  reconstruction (see next Section).

According to the above discussion, the expectation is that: (i) For  $\lambda$  too small one will obtain rough parameter  $\alpha$  variations (highly correlated to the residual errors in step 1) and, consequently, large  $E_{\text{smooth}}$  and very small  $E_{\text{mod}}$  values (the inputs  $\alpha_t$  will be tuned to produce outputs of the ANN that exactly match the targets). (ii) For  $\lambda$  too large  $E_{\text{smooth}}$  will be almost zero,  $\alpha_t$  will be nearly constant and, consequently,  $E_{\text{mod}}$  will not vary much from the values obtained in step 1. (iii) For intermediate values of  $\lambda$  there should be a region in which in  $E_{\text{smooth}}$  becomes almost insensitive to  $\lambda$ . The curves  $\alpha_t$  vs.  $t$  in this region will be very similar to each other

and any one of them can be taken as a quasi-optimal reconstruction.

For the backpropagation rule, the adjustment of the input  $\alpha_t$  requires the calculation of the gradients<sup>10</sup>  $\partial E_{\text{smooth}}/\partial \alpha_t = 2(\alpha_{t+1} + \alpha_{t-1} - 2\alpha_t)$  and  $\partial E_{\text{mod}}/\partial \alpha_t = (x_{t+1} - o_{t+1}) \sum_k w_{ok} \sigma_k (1 - \sigma_k) w_{k\alpha}$ . Here  $k$  runs on hidden-layer units,  $w_{ok}$  are weights and biases from these units to the output one, and  $w_{k\alpha}$  are the connections between the extra neuron for the  $\alpha$  input and the hidden units. As usual,  $\sigma$  denotes the sigmoid activation function. Notice that by a rescaling of weights and shift of biases it is always possible to choose  $\alpha_t$  so that its mean value  $E[\alpha] \equiv 0$  and its variance  $E[\alpha^2] - E^2[\alpha] \equiv 1$ , which implies that the dynamics perturbation can only be reconstructed up to a linear transformation. In step 2 we have initialized  $\alpha_t = 0$  and enforced these two conditions after each training epoch to speed up the learning process and to set an absolute scale for  $\lambda$ . Finally, we performed step 1 to help avoiding the wrong tuning of the  $\alpha$  inputs to the initial errors the ANN makes when it starts learning the targets.

### 3. Applications

#### 3.1. Synthetic data

We have implemented the strategy described in Section 2 in a controlled situation, which allowed us both to validate the proposed algorithm and to gain a deeper understanding on the problem. We considered the chaotic logistic map with additive noise:  $y_{t+1} = ry_t(1 - y_t)$ ,  $x_{t+1} = y_{t+1} + \varepsilon_{t+1}$ , where  $\varepsilon_t$  is Gaussian noise corresponding to a noise-to-signal ratio of  $10^{-3}$ . The parameter  $r$  was slowly driven according to the law  $r_t = r_0 - A \cos(2\pi t/T) \exp(-t/T)$ , with  $r_0 = 3.8$ ,  $A = 0.045$  and  $T = 50$  (see Fig. 2), which keeps the map in the interesting chaotic regime. Then, the efficacy of the algorithm proposed above can be assessed by considering the reconstruction error

$$E_{\text{rec}} = \frac{1}{N\sigma_\alpha^2} \sum_{t=1}^N (r_t - r_0 - \alpha_t)^2. \quad (4)$$

According to its definition,  $E_{\text{rec}} = 0$  for a perfect reconstruction  $\alpha_t = r_t - r_0$  and  $E_{\text{rec}} = 1$  for constant  $\alpha_t = E[r - r_0] \simeq 0$ . We considered only  $N=100$  iterates in the data set  $D$  and, unlike what we did in Ref. 10 where we used all the available data for training the ANN, here we randomly split these iterates

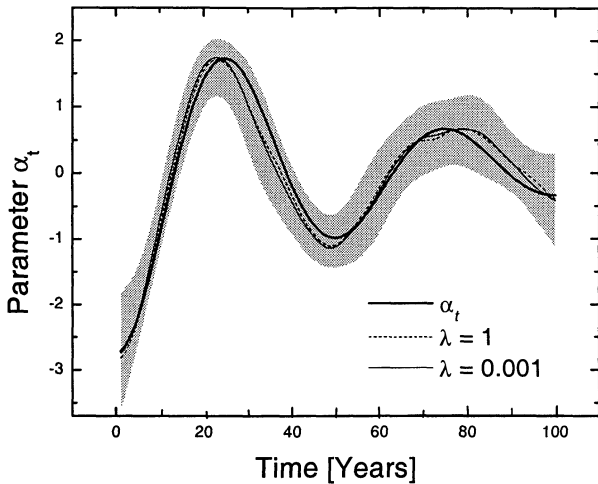


Fig. 2. Forcing parameter  $\alpha$  for the logistic map (thick full line) and its best reconstructions for  $\lambda = 0.001$  (thin full line) and  $\lambda = 1$  (dashed line). The gray area indicates the  $\pm 1.96\sigma$  dispersion of 50 independent runs of our algorithm.

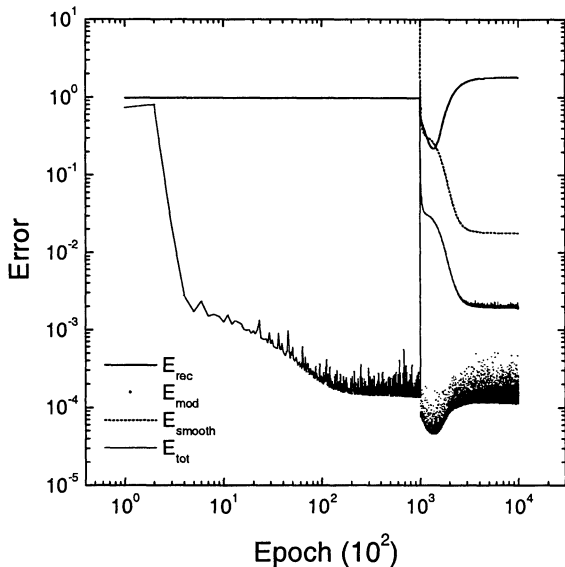


Fig. 3. Evolution of different errors during the training process for the case of the forced logistic map and  $\lambda = 0.1$ .

in 80 points for learning and 20 points for validating the algorithm. For modelling purposes we used a simple 2:3:1 ANN trained with the standard back-propagation rule. Fig. 3 shows typical evolutions of the different errors as a function of the number of training epochs. The errors' behaviors are very similar to what we found in Ref. 12, and confirm the

intuitions in Sec. 2: (i) From epoch 1 to  $10^4$  we set  $\alpha = 0$ . The total error  $E_{\text{tot}}$  starts decreasing during training, until it settles around epoch  $2 \times 10^4$  in a relatively small error value. Then, at epoch  $10^5$  we switch on the extra input and  $E_{\text{tot}}$  jumps to a large value that depends on  $\lambda$  ( $=0.1$  in this case); it then rapidly decreases, showing first a shoulder and finally reaching a plateau after epoch  $1.2 \times 10^5$ . Here the training process has forced a crossover to a final situation in which  $\alpha$  becomes too smooth. (ii) Until epoch  $10^5$   $E_{\text{mod}}$  is identical to  $E_{\text{tot}}$ . At this epoch the network starts making use of the new input to model the dynamics, and  $E_{\text{mod}}$  drops almost one order of magnitude. Finally, when  $\alpha$  becomes too smooth it increases up to a value slightly below the minimum error without the extra input (the final reconstructed perturbation for  $\lambda = 0.1$ , even if not completely correct, still helps in modelling the nonstationary dynamics). (iii)  $E_{\text{smooth}}$  is only defined after epoch  $10^5$ . It starts from a large value and quickly decreases, having a behavior very much like that of  $E_{\text{tot}}$ . For the chosen value  $\lambda = 0.1$  the training process smoothes  $\alpha$  too much, and the final plateau in  $E_{\text{smooth}}$  corresponds to this situation. (iv) The error  $E_{\text{rec}}$  in the reconstruction of  $\alpha$ , only available in this controlled situation, has the expected behavior: Until epoch  $10^5$  it is equal to one by construction (since we initialize  $\alpha_t = 0$ ). Then, while the algorithm seeks the appropriate parameter variation,  $E_{\text{rec}}$  drops almost one order of magnitude, until approximately at epoch  $1.4 \times 10^5$  it starts increasing because of the excessive smoothing of  $\alpha$  discussed above.

The errors' evolutions in Fig. 3 are characteristic for intermediate to large values of  $\lambda$ . The most interesting conclusion that can be drawn from them is the fact that the behaviors of  $E_{\text{mod}}$  and  $E_{\text{rec}}$  as functions of the training epochs are very similar. To some extent this was to be expected, since good reconstructions of  $\alpha$  must lead to small modelling errors and vice-versa. Therefore, as a practical criterion to identify the best reconstruction for each value of  $\lambda$  in a real situation (where  $E_{\text{rec}}$  is not available), in<sup>10</sup> we considered the “optimal”  $\alpha$  values to be the corresponding ones at the minimum of  $E_{\text{mod}}$ . We will see below how this criterion can be validated and perfected by considering the behavior of the generalization error on the 20 points set apart from the training data.

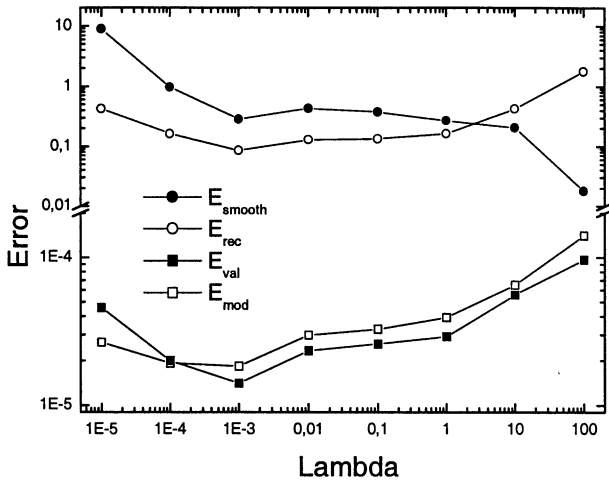


Fig. 4. Different errors at the minimum of  $E_{\text{mod}}$  as a function of  $\lambda$  for the forced logistic map.

In order to explore the validity of the criterion used in Ref. 10, we plot in Fig. 4 all the errors at the minimum of  $E_{\text{mod}}$  as functions of  $\lambda$ . For instance, the values for  $\lambda = 0.1$  shown in this figure correspond to the network trained approximately  $1.4 \times 10^5$  epochs (see Fig. 3). Actually, the results in Fig. 4 are average values obtained by performing 100 independent experiments for each value of  $\lambda$  (starting from the random splitting of the data in training and validation sets) and selecting the best 50 networks having the smallest prediction error  $E_{\text{val}}$  on the validation set. The reconstructed signal for predicting point  $i$  in this set was taken to be  $\alpha_i = (\alpha_{i-1} + \alpha_{i+1})/2$ , that is, the mean of the values for the nearest-neighbor points ( $i-1, i+1$ ) in the training set (which should be a good approximation because of the smooth  $\alpha$  behavior assumption). In this way we reduce the possibility of having badly trained ANNs, something that in Ref. 10 was done by keeping only the network with the smallest  $E_{\text{mod}}$  in 10 independent experiments. The method followed here is more robust and will lead to a more accurate criterion to select the optimal  $\lambda$  value, as described below.

Let's discuss in detail the errors' behavior as a function of  $\lambda$ :

- $E_{\text{smooth}}$  starts from a large value at small  $\lambda$ , and decreases with increasing  $\lambda$  until it reaches an extended plateau around  $\lambda \sim 10^{-3}$ . In the region  $10^{-3} \leq \lambda \leq 1$ ,  $E_{\text{smooth}}$  is practically insensitive to  $\lambda$  (by comparison with the orders of magnitude changes outside this region), which we will con-

sider a signature of good  $\alpha$  reconstructions. Then, for  $\lambda \sim 10$   $E_{\text{smooth}}$  has a sudden drop, indicating that  $\alpha$  becomes too smooth.

- In accordance with the behavior of  $E_{\text{smooth}}$ , we see that the reconstruction error  $E_{\text{rec}}$  shows similar quality for  $\alpha$  curves obtained with  $\lambda$  values that, remarkably, may differ up to 3 or 4 orders of magnitude.
- $E_{\text{mod}}$  essentially shows a rise with  $\lambda$  that was to be expected: In the small  $\lambda$  region the network tunes the extra input to cancel the modelling errors, without paying much attention to the term  $\lambda E_{\text{smooth}}$  since its value is negligible with respect to  $E_{\text{mod}}$ . On the contrary, for large  $\lambda$  the parameter variation becomes too smooth to reduce as much as possible  $E_{\text{smooth}}$ , and  $E_{\text{mod}}$  tends to the value obtained without the extra input. In between, there is an uninteresting increase of this latter error.
- We have also plotted the average prediction errors  $E_{\text{val}}$  on the validation sets. Interestingly enough, we see that the minimum of  $E_{\text{val}}$  coincides with the minimum of  $E_{\text{rec}}$ . This is natural since one expects that the best  $\alpha$  will lead to the most accurate modelling of the dynamics, thus producing the best predictions on unseen data. This can be taken as a more accurate criterion for determining the optimal  $\lambda$  value. In Ref. 10 we had used an *ad hoc* criterion, choosing the largest  $\lambda$  in the  $E_{\text{smooth}}$  plateau to have the smoothest possible curve. In passing, notice that the behavior of  $E_{\text{mod}}$  seems also to indicate the optimal reconstruction of  $\alpha$ , but we do not have an explanation for this.

In Fig. 2 we present the best reconstruction of  $\alpha$ , corresponding to an average over the 50 curves obtained at  $\lambda = 10^{-3}$ , together with the  $\pm 1.96\sigma$  error bounds estimated from the standard deviation in these experiments. For this average  $\alpha$  the corresponding  $E_{\text{rec}} = 0.035$ , which should be compared with  $E_{\text{rec}} = 0.041$  obtained with the method proposed in Ref. 9. However, in this last case the error is measured only on 80 points of the record, since we cannot reconstruct the perturbing signal in the first and last 10 points. This is so because the algorithm in Ref. 9 requires using a moving window, which for this problem we chose to be of length 20 points. Another advantage of the present approach is that the

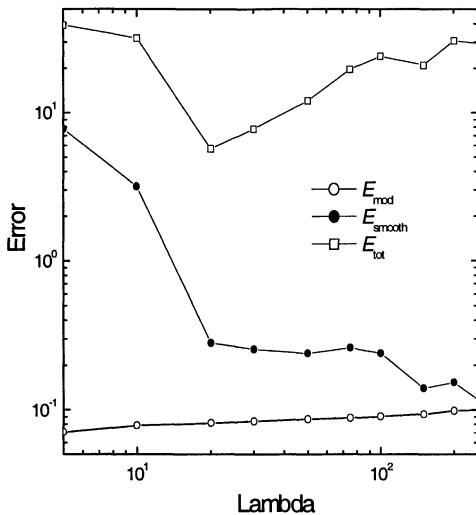


Fig. 5. Different errors at the minimum of  $E_{\text{mod}}$  as a function of  $\lambda$  for the sunspot time series.

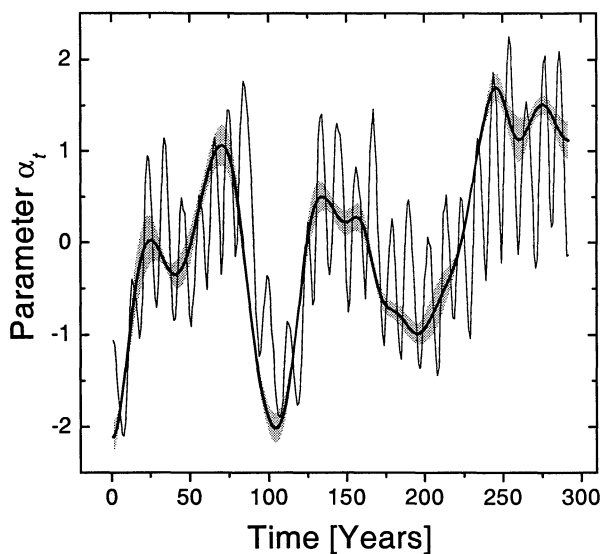


Fig. 6. Reconstruction of  $\alpha$  for the sunspot problem (thick line). The gray area indicates the  $\pm\sigma$  dispersion of 10 independent runs of our algorithm. The thin line gives the evolution of the Sun's large-scale magnetic field from.<sup>13</sup>

ANN simultaneously reconstructs  $\alpha$  and models the intrinsic dynamics. For comparison, we also show the average  $\alpha$  reconstructed with  $\lambda = 1$ , at the other end of the  $E_{\text{smooth}}$  plateau. Remarkably, for this value of  $\lambda$ , three orders of magnitude larger than the optimal one, the reconstructed curve is hardly distinguishable from the previously obtained  $\alpha_t$ .

### 3.2. Real-world data

We briefly rediscuss here the results obtained in Ref. 10 for the sunspot time series. This is done mainly to allow introducing an important question for this application — in fact, for any real-world application — which was not discussed in our previous work: The identification of the reconstructed signal. The sunspot record is a difficult problem since it is very noisy and has a complex dynamics with several time scales.<sup>11</sup> As pointed out in Ref. 10, a thorough investigation would require optimizing the ANN architecture used and controlling the possibility of overfitting. In this preliminary study we worked instead with a very small 3:2:1 ANN, which is not optimal in terms of modelling error but it should not lead to serious overfitting problems. In Fig. 5 we plot the errors at the minimum of  $E_{\text{mod}}$  as a function of  $\lambda$ , using the whole data set for modelling purposes and choosing the ANN with the smallest  $E_{\text{mod}}$  in 10 different experiments. The main difference with the logistic map is the reduction of the  $E_{\text{smooth}}$  plateau, although  $\alpha$  is still roughly independent of  $\lambda$  for nearly a decade (20 ~ 150). Fig. 6 shows the reconstructed  $\alpha$  for  $\lambda = 100$ , and the bounds obtained from the 10 experiments. The two main features observed in this figure are the periodic structure of approximately 100-year period corresponding to the Gleissberg cycles<sup>11</sup> and the important rise in amplitude in the last cycle. Taking into account the recently-estimated doubling in the Sun's coronal magnetic field in the last century,<sup>12</sup> this last effect suggests a connection between  $\alpha$  and the solar magnetic field. To check this conjecture, we also show in Fig. 6 the secular evolution of the Sun's large-scale (open) flux.<sup>13</sup> Its strong overall similarity with  $\alpha$  (disregarding the short solar cycle oscillations with a period of approximately 11 years) indicates a close relationship of the hidden driving signal and the Sun's magnetic field. In order to explore more in depth this connection and answering other interesting questions, a thorough investigation of the sunspot time series is currently being performed.

### 4. Conclusions

We have refined and extended the ANN-based method for the reconstruction of perturbing signals from nonstationary time series proposed in Ref. 10. The algorithm simply incorporates an extra input

unit in a feedforward ANN, and adjusts the corresponding input values during the training phase to learn simultaneously the intrinsic dynamics and the temporal profile of the driving parameter. This is achieved by minimizing a suitable-defined error function, depending on a hyperparameter  $\lambda$ . We found here that the use of validation data, held out from the training set, allows us to give an accurate criterion to select the optimal value of this parameter. In addition, from a controlled example corresponding to a forced logistic map, we showed that our algorithm outperforms previous methods in the literature, allowing a better modelling of the nonstationary system. Moreover, a preliminary application to the real-world sunspot time series uncover secular changes in solar dynamics and links the hidden perturbing signal to the evolution of the solar magnetic field.

Several extensions of the algorithm here proposed are possible: tracing the simultaneous variation of several parameters, consideration of general regression problems where the underlying phenomenon is changing in time, etc. These and other applications are currently under investigation.

### Acknowledgments

This work was partially funded by ANPCyT of Argentina through PICT 11-03834.

### References

1. H. Kantz and T. Schreiber 1997, *Nonlinear Time Series Analysis*, Cambridge Nonlinear Science Series 7 (Cambridge Univ. Press, Cambridge).
2. M. Casdagli 1997, "Recurrence plots revisited," *Physica D* **108**, 12–44, and references therein.
3. T. Schreiber 1999, "Interdisciplinary application of nonlinear time series methods," *Phys. Rep.* **308**, 1–64.
4. R. Manuca and R. Savit 1996, "Stationarity and nonstationarity in time series analysis," *Physica D* **99**, 134–161; T. Schreiber 1997, "Detecting and analysing nonstationarity in a time series using nonlinear cross predictions," *Phys. Rev. Lett.* **78**, 843–846.
5. F. Lombard and J. D. Hart 1994, "The analysis of change-point data with dependent errors, in *Change-Point Problems*," eds. E. Carlstein, H.-G. Muller and D. Siegmund, IMS Lecture Notes — Monograph Series, Vol. 23 (Inst. Nath. Statist., Hayward, CA); M. B. Kennel 1997, "Statistical test for dynamical nonstationarity in observed time-series data," *Phys. Rev.* **E56**, p. 316.
6. J. Stark 1999, *J. Nonlinear Sci.* **9**, p. 255; R. Hegger, H. Kantz, L. Matassini and T. Schreiber 2000, *Phys. Rev. Lett.* **84**, 4092–4095.
7. J. Stark and B. V. Arumugan 1992, "Extracting slowly varying signals from a chaotic background," *Int. J. Bifurcation and Chaos* **2**, 413–419; L. M. Hively, P. C. Gailey and V. A. Protopopescu 1999, *Phys. Lett.* **A258**, p. 103.
8. K. M. Short 1997, "Signal extraction from chaotic communications," *Int. J. Bifurcation and Chaos* **7**, 1579–1597.
9. P. F. Verdes, P. M. Granitto, H. D. Navone and H. A. Ceccatto 2001, "Nonstationary time-series analysis: Accurate reconstruction of driving forces," *Phys. Rev. Lett.* **87**, 124101–124104.
10. M. I. Szliga, P. F. Verdes, P. M. Granitto and H. A. Ceccatto, *Extracting Driving Signals from Nonstationary Time Series*, VII Brazilian Symposium on Artificial Neural Networks (SBRN'2002) (Porto de Galinhas, Recife, November 11–14), pp. 101–105.
11. W. Gleissberg 1967, *Solar Phys.* **2**, p. 231; A. Garca and Z. Mouradian 1998, *Solar Phys.* **180**, p. 495.
12. M. Lockwood, R. Stamper and M. N. Wild 1999, "A doubling of the Sun's coronal magnetic field during the past 100 years," *Nature* **399**, 437–439.
13. S. K. Solanki, M. Schuessler and M. Fligge 2000, "Evolution of the Sun's large-scale magnetic field since the Maunder Minimum," *Nature* **408**, 445–447.

