Building a Neuro-Fuzzy System to Efficiently Forecast Chaotic Time Series

Léonard Studer [1] and Francesco Masulli

Istituto Nazionale per la Fisica della Materia,
Dipartimento di Fisica, Universitá di Genova,
via Dodecaneso 33, I-16146 Genova, Italia
E-mail: {studer|masulli}@ge.INFM.it

In this paper we show which elements have to be extracted from a chaotic time series in order to define the architecture of a forecaster. The forecaster chosen here is a Neuro-Fuzzy System. This NFS is trained by a supervised gradient descent algorithm. The NFS is made of a layer of singleton inputs, a hidden layer of Gaussian memberships functions and one output unit. Product is used for rule inference and sum for rule composition. Output is given by a height defuzzifier. Test cases based on Mackey–Glass time series are presented.

Key words: Time Series, Forecasting, Chaos, Fuzzy Logic, Artificial Neural Networks
PACS: 05.45.+b, 07.05.Mh

1 Introduction

For about 20 years now, time series with chaotic behavior are regularly observed. Chaos has been seen in many natural phenomena: meteorology, near turbulent fluids, cardiology, climatology to cite a few.

The identification of such a behavior has important consequences either in the scientific understanding of the phenomenon or in the technical mastering of the studied system. This identification step relies on a set of rather new mathematical and algorithmic tools that have been described in the published scientific literature during the last 2 decades.

[1] Supported by the Swiss National Science Foundation

Assuming that the identification step has been successfully achieved, one can consider to actually use the chaotic behavior to advantage. For example, the dynamic stabilization of a chaotic system or the transmission of a signal hidden in a chaotic wave have been suggested and experimentally shown.

The control of the chaotic system requests its modeling. Here, either the dynamical equations are known or the system has to be approximated by some general mathematical tools. Of course, dynamical equations are known only for the simplest systems. Usually, one has to rely on non-specific approximators such as Artificial Neural Networks. In the study we report here, we used a Neuro-Fuzzy System to carry on the modeling task.

We have attacked the problem of forecasting meteorological time series (exactly sea level atmospheric pressure waves). For that purpose, we have set up a Neuro-Fuzzy System (NFS) that has demonstrated good capacities in forecasting this natural phenomenon[1].

A noteworthy property of the NFS we used is that it has been proved to be an *Universal Approximator*[2] on any real continuous function on a compact support to an arbitrary precision. Multi-Layer Perceptrons and Radial Basis Functions are also of this kind.

Some other advantages to use a NFS deserve to be mentioned: 1) The possibility to easily input some *a priori* knowledge (from an expert) to bias the NFS towards the problem to be solved. 2) By training, NFS can learn internal relations in numerical data sets. 3) Extraction of learned rules by a trained NFS is possible in form of meaningful linguistic (fuzzy) relations.

In our case, a chaos-inspired analysis of the time series is a source of knowledge that can be used to shape the NFS. We will show exactly how we incorporate some of the results of the analysis in our NFS.

Results obtained on a synthetic chaotic Mackey–Glass time series will be shown and prospects we are currently studying will be given in conclusion.

2 Fuzzy Logic Systems

Fuzzy set theory[3] aims to capture the sort of uncertainty related to ambiguity in classifying objects. Relations between fuzzy sets is the topic of fuzzy logic.

Although the word "fuzzy" could be a misnomer, the theory is rigorously developed. Also, differences and similitude between fuzzy set theory, probability and other uncertainty theory (e.g. evidence theory) have been vividly debated for years and carefully studied. For more detailed information on the fuzzy theme, see the good textbook of Klir and Yuan[4].

In the following two subsections, we give some ideas on fuzzy sets and fuzzy logic systems and then we describe the NFS in detail.

2.1 Fuzzy Sets and Fuzzy Systems

Fuzzy set theory is an extension of the conventional (crisp) set theory. A *fuzzy set* A is defined via a *membership function* $\mu_A(x)$ which gives the membership grades of elements x to the fuzzy set A. By construction, fuzzy sets are very convenient to numerically capture linguistic concepts such as "large", "warm", or "cold".

The form of the membership function is arbitrary and has to be determined in function of the problem. Methods have been devised for this determination, inspired from knowledge engineering or statistics. Neuro-Fuzzy Systems had been created for that purpose.

Fuzzy logic relates these linguistic variables (i.e. fuzzy sets) through operations. Fuzzy logical operations between fuzzy sets are extensions of classical connectives such as intersection, union, set-complement, AND, OR, THEN etc.

Fuzzy logical operations are not unique. For one given fuzzy logical operation (let say intersection) there is an infinite family of usable operations. Fuzzy intersection can be represented by *min* or *product*, fuzzy union can be *max*, or *sum* operations (among others). Fuzzy complement is approprietely represented by complement to 1 of the membership function. As in classical logic, logical conjunctive operation (AND) is implemented by an intersection between sets; disjunction by union and negation by set-complement.

Fuzzy number, fuzzy arithmetic and fuzzy calculus have also been defined, see the cited literature for more information.

Fuzzy Systems are constituted by four components:

- the *fuzzification module* that transforms the *crisp* (i.e. not-fuzzy) input data coming from the real world into membership values;
- the *fuzzy rule base* with a bank of *fuzzy if-then rules* or *fuzzy conditional statements*, of the type:

  $$\textbf{IF } x_1 \in A \textbf{ AND } x_2 \in B \textbf{ THEN } y \in C,$$

  where A, B and C are fuzzy sets and $x_1 \in A$ is given by $\mu_A(x_1)$;

- the *decision making unit* or *fuzzy inference engine* performing the inferences on the rules following the selected approximate reasoning method;
- the *defuzzification module* that transforms the fuzzy sets resulting from fuzzy inference into crisp outputs.

2.2 The Neuro-Fuzzy System

The NFS[2] we used here is a fuzzy system based on the following assumptions: height defuzzification, sum composition, product inference rules, Gaussian membership functions, and singleton fuzzifier. The NFS can be associated to a feedforward connectionist system with only one hidden layer. More specifically, if there are K units in the input layer, J fuzzy inference rules and I outputs, the rule activations can be written as:

$$r_j = \prod_k \mu_{jk}(x_k). \tag{1}$$

The quantity $\mu_{jk}(x_k)$ is the value of the membership function of the component x_k of the input vector for the j-th rule, and is defined as:

$$\mu_{jk}(x_k) = \exp\left(-\frac{(x_k - m_{jk})^2}{2\sigma_{jk}^2}\right), \tag{2}$$

where m_{jk} and σ_{jk}^2 are the means and the variances.

The values of the output units are:

$$y_i = \frac{\sum_j r_j s_{ij}}{\sum_j r_j} \tag{3}$$

and s_{ij} is the fuzzy singleton of the j-th rule associated with the output y_i.

In our experiments, no *a priori* knowledge has been used to enforce (before the learning phase) some fuzzy if-then rules in the NFS. We obtained the NFS parameters (i.e., m_{jk}, σ_{jk} and s_{ij}) by performing a gradient descent across the training set with respect to the mean square error (MSE). The formula for the gradient descent can be found in literature[2].

3 Elements from chaos theory

Chaotic trajectories evolve in a phase space of dimension N, with dynamical coordinates $x, \dot{x}, y, \dot{y}, z, \dot{z} \ldots$

After any initial transients have died out, trajectories evolve on a substructure of the complete phase space, the *strange attractor*. This geometric object is in general of fractal (or box counting) dimension D.

Usually, chaotic system are measured by a scalar time series $(x_1, x_2, x_3 \ldots x_n)$ of *one* of the N dynamical coordinates. A time delayed vector $\mathbf{v}$ of dimension m can be reconstructed from this time series: $\mathbf{v} = (x_1, x_2 \ldots x_m)^\top$. By virtue of the Takens–Mañé theorem[5], the trajectory of the vector $\mathbf{v}$ will be the result of a diffeomorphic transformation (applied on the true state point) that keeps the important relevant properties of the system; in particular, one reconstructed point $\mathbf{v}_0$ can not be the source of 2 different trajectories (Laplacian determinism).

The Takens–Mañé theorem states that $m > 2D$ is a sufficient condition to achieve such a "fair" reconstruction. Note that $m \in [N, D]$ could also lead to a "fair" reconstruction, but this can not be demonstrated for the general case.

Of course, in forecasting we are very interested to achieve this "fair" reconstruction and it is why the result of the TM theorem ($m > 2D$) has to be included in our NFS forecaster. From the TM theorem, the way is now easy: we need a number of input nodes d greater than $2D$.

The determination of the requested number of inputs d is linked to the determination of the box counting dimension D. This can be achieved by other algorithmic methods.

Also, our system, being chaotic, has a limited time-horizon in forecasting. Any inaccuracy in the initial state vector will be amplified during the dynamical evolution up to the point to prevent any useful predictions. This practical limit in time implies that the time delay τ between two scalar measurements x_i and x_{i+1} should not be too big. If τ is too long, x_i and x_{i+1} will be essentially unrelated, beyond the time horizon. If τ is chosen too short, x_{i+1} will be a "copy" of x_i. Chaos literature gives some prescriptions on how to chose the optimal τ. But these prescriptions are not grounded in theory.

4 Some experimental results

To shed more lights on the optimal values for m and τ, we have chosen to use a synthetic time series based on the Mackey–Glass equation: $\dot{x}(t) = ax(t-\Delta)/\left(1+x(t-\Delta)^c\right) - bx(t)$ with $a = 0.2$, $b = 0.1$ and $c = 10$. Varying Δ from 17 to 100 let vary D from 2.1 to 10 ([6]). This feature is very useful, because, we can visualize low dimensional chaos and on the high dimension side, the MG system is closer to the meteorological data we studied in. We use MG time series with $\Delta = 17$ ($D_{17} \approx 2.1$ and $m_{17} \geq 5$) and $\Delta = 30$ ($D_{30} \approx 3.6$ and $m_{30} \geq 8$).

We have trained our NFS with a training set of 6000 patterns $\mathbf{v}_t$ and we have used a test set of 1000 vectors. Initial values of the parameters to be adjusted were set randomly. Convergence of the learning phase is fast. By example, the generalization normalized error is less than 1% after only 10 training epochs, for a 4/8/1 network, which is quite good. The training procedure was left running for 500 epochs, where usually the MSE was still decreasing but in small proportion. No sign of overfitting was observed.

The configuration of the NFS was d input nodes, $2d$ hidden nodes and 1 output node, while d scanned the range 2 to 15. Time delay τ between measurements was also tested for value between 1 and 24.

In fig. 1, a sharp diminution of $\log(error)$ can be seen, as soon as d is big enough. About τ, no clear value seems optimal for low forecasting errors: apparently, best results are achieved for short τ. Indication from chaos literature would have suggested a value of τ of about 11 (the first minimum of the average mutual information of the time series).

We have also added some noise to the clean time series. Next, we have trained, on the noisy time series, a NFS forecaster whose number of inputs was minimal for the clean series. We have then noted a degradation of the forecasting performance. This is expected, as a pure stochastic noise is seen — in chaos theory — as a dynamical system whose trajectory evolves in a phase space of infinite dimension. So, the dimension of the attractor of the noisy series is effectively higher than for the clean case. To recover good forecasting performances, we can either add new inputs to the NFS (which will imply longer training time) or filter the noisy signal. But because chaotic time series have broadband frequency spectra, we have to used non-linear filtering methods. Among various possibilities, let mention that Wang[2] described how to use a NFS as a non-linear filter!

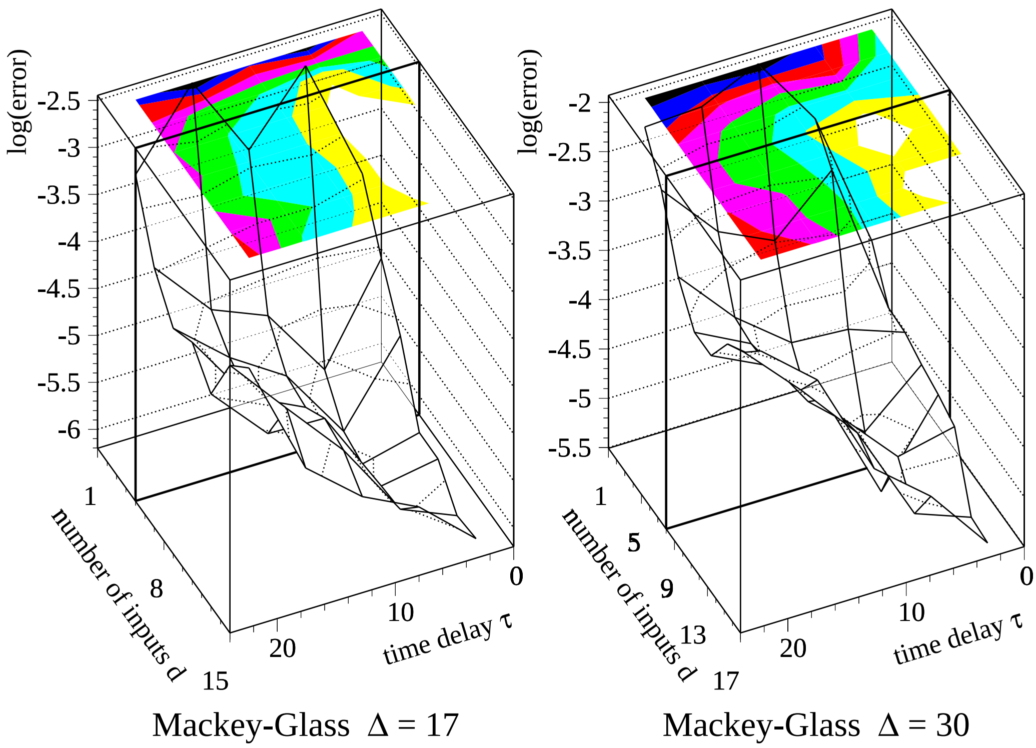

Fig. 1. Dependence of the forecasting error in function of the number d of inputs and of the time delay τ. Takens–Mañé theorem requires $d \geq 5$ (left) or $d \geq 8$ (right).

5 Conclusions and Prospects

We have shown that the use of a Neuro-Fuzzy System for forecasting time series is promising. Learning time and efficiencies are quite good. Moreover, NFS has the key property to be an universal approximator. This gives a strong mathematical ground applying NFS in time series forecasting.

Efficiency in accurate and robust forecasting can not rest solely on a good algorithm. We demonstrated that the NFS has to include proper features of the time series. In this study, we have used the Takens–Mañé theorem. The twists, branching and foldings of the strange attractor are also relevant features of the chaotic system. Works in chaos theory[7] show that the related topological invariants are quite robust against perturbations (ext. noise and/or variation of the chaos control parameter).

How do we envision to use these robust topological invariants in our NFS ? In a very recent preliminary study, we have presented the set of the reconstructed time delayed vectors $\mathbf{v}_t$ to an unsupervised growing neural gas (GNG)[8]. At the end of this learning phase, the GNG nodes cover the m-dimensional phase space of the strange attractor and each node is associated to a localized part of the attractor. By construction, the number of links from one node is equal to the local (integer) dimension of the attractor. We have also noted that the mean number of links of the GNG was proportional to the global dimension D of the attractor. These results show that (a part of) the topology of the strange attractor is learned by the GNG. Transferring the structure of the GNG to the hidden layers of the NFS will then incorporate (a part of) the topology of the attractor into our NFS-forecaster.

Acknowledgement

Original source code of the Neuro-Fuzzy System used here was written by F. Casalino and R. Caviglia from the University of Genoa.

References

[1] F. Masulli, F. Casalino, R. Caviglia, L. Papa, "Comparison of statistical methods and fuzzy systems in atmospheric pressure wave prediction (invited paper)", in: *Appl. and Science of Art. Neural Net. - SPIE Proc. Series* (ed. S. K. Roger and D. W. Ruck, publ. SPIE - Bellingam, WA, USA) **2492** (1995) 1050–1061

[2] L. X. Wang, *Adapive Fuzzy Systems and Control* (publ. Prentice Hall, 1994)

[3] L. A. Zadeh, "Fuzzy sets", *Information and Control* **8** (1965) 338–352

[4] G. J. Klir and Bo Yuan, *Fuzzy Sets and Fuzzy Logic* (publ. Prentice Hall, 1995).

[5] T. Sauer, J. A. Yorke, M. Casdagli, "Embedology", *Journ. of Stat. Physics* **65 3–4** (1991) 579–616

[6] J. D. Farmer, "Chaotic attractors of an infinite-dimensional system", *Physica D* 4 (1982) 366–393

[7] N. B. Tuffilaro, "Topological organization of (low-dimensional) chaos", in: *From statistical physics to statistical inference and back* (ed. J.-P. Nadal and P. Grassberger, NATO-ASI Proceedings, publ. Kluwer Academic, 1994)

[8] M. Kunze, "The Neural Networks Objects", in: this issue of NIM (1996)