ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 157-164

Times Series and Neural Network : a Statistical Method for Weight Elimination

M.Cottrell, B.Girard, Y.Girard, M.Mangeas

Centre de Recherche SAMOS
Université Paris 1, 90 rue de Tolbiac, 75634 PARIS Cedex 13

Abstract. *Many authors use feedforward neural networks for modelling and forecasting time series. Most of these applications are mainly experimental and it is often difficult to extract a general methodology of the published studies. In particular, the choice of the architecture is a tricky problem. We try to combine the statistical techniques of linear and nonlinear time series with the connectionist approach. The asymptotical properties of the estimators lead us to propose a systematical methodology to determine which weights are nonsignificant and to eliminate them in order to simplify the architecture. This method is applied to the famous Sunspots benchmark and to some artificial series.*

1 Introduction[1]

We are interested in time series processes, which can be viewed as generalized non-linear ARMA models, defined by a recurrence equation

$$Y_t = f(Y_{t-1}, Y_{t-2}, \ldots, Y_{t-p}, X_t, W) + \epsilon_t \text{ , for } t \in \mathbb{Z} \qquad (1)$$

where (X_t) is a $\mathbb{R}^q$ valued sequence (deterministic or not sequence)
f is a (generally non linear) function : $\mathbb{R}^{p+q+m} \rightarrow \mathbb{R}$,
the vector W is the parameter ($\in \mathbb{R}^m$) of the model,
(ϵ_t) is a sequence of independent and identically distributed random variables, with mean 0 and finite variance σ^2, such that ϵ_t is independent from the past $(Y_s)_{s<t}$.

The equation (1) defines a very large and general class of models. *The idea is to restrict ourselves to the class of the functions f which are associated*

[1]This work was partially supported by EDF, Direction des Etudes et des Recherches, Clamart.

ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 157-164

with a multilayer perceptron, with one hidden non-linear layer, one input layer with $(Y_{t-1}, Y_{t-2}, \ldots, Y_{t-p}, X_t)$ *as inputs, and one linear output unit whose desired value is* Y_t.

This choice of the function f is motivated by recent but now classical results of the approximation capability of multilayer perceptrons. ([Cybe], [Horn(a)]),[Horn-(b)].

Many recently published studies (for example : [Ginz], [Lape], [Varf], and others) use neural networks for time series analysis, but most of them are overall experimental. Other papers propose more general methodologies : for example, C.de Groot and D.Würtz [Groo] present an exhaustive comparison between various statistical and connectionist analysis applied to two examples, A.S.Weigend et al. [Weig] introduce an intuitive method for weight elimination, etc... Here we take a theoretical approach, based on prior statistical knowledge, and in particular on the asymptotical properties of the weight estimators.

2 The Linear Case

Let (Y_t) be a linear autoregressive process AR(p), represented by the recurrence equation

$$Y_t = \mu + \sum_{i=1}^{p} \alpha_i Y_{t-i} + \epsilon_t \qquad (2)$$

for $t \in \mathbb{Z}$, and where some coefficients α_i can be constrained to be 0.

The assumptions are :

1) The process (Y_t) is a 2^{nd} order stationary process, i.e. $E(Y_t)$ is a constant and $Cov(Y_t, Y_{t+h}) = \gamma_h$ depends only on h, and not on t.

2) (ϵ_t) is a sequence of independent centered random variables with a *common distribution* and finite variance σ^2,

3) ϵ_t is independent of the past of the process (Y_t) (which is equivalent to the condition :

The roots of the polynomial $\sum_{i=0}^{p} \alpha_i \, z^{p-i} = 0$, with $\alpha_0 = -1$, lie inside the unit circle).

Note that we do not consider in this paper the case of the autoregressive moving-average models, (ARMA models), which can be written

$$Y_t = \mu + \sum_{i=1}^{p} \alpha_i Y_{t-i} + \sum_{j=0}^{q} \alpha'_j \, \epsilon_{t-j} \ .$$

It is clear that (2) can be implemented by a linear neural network (ADALINE), with p inputs, one input fixed to 1 for the threshold μ, no

ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 157-164

hidden layer, and a single output unit. The coefficients $(\alpha_1, \alpha_2, \ldots, \alpha_p, \mu)$ are the connection weights.

Using $T+p$ observations $(Y_1, Y_2, \ldots, Y_{T+p})$, learning in the network consists of minimizing the sum of the squared residuals

$$S(\alpha, \mu) = \Sigma(Y_t - \hat{Y}_t)^2,$$

where $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_p)'$ (the $'$ is the transposition operator) and

$$\hat{Y}_t = \mu + \sum_{i=1}^{p} \alpha_i Y_{t-i}\,.$$

This is exactly the Box-Jenkins conditional least squared method [Box]. It is refered to as conditional, since the method depends on the first p values of the process (Y_t).

This model can be generalized to an autoregressive processes with *exogenous variables.* In this case, one has to add as many input units as there are exogenous variables.

The estimators $\hat{\alpha}$ of the parameters are the least squared estimators, whose properties are well-known : given the assumptions 1)-3), one can prove that the $\hat{\alpha}$ are asymptotically Gaussian. One has

$$\sqrt{T}(\hat{\alpha} - \alpha) \xrightarrow{\mathcal{D}} \mathcal{N}_p(0, \sigma^2 \Sigma^{-1})$$

where T tends to infinity, $\sigma^2 = Var(\epsilon_t)$, and Σ is the $(p \times p)$ autocovariance-matrix defined by $(\Sigma)_{ij} = Cov(Y_{t+i}, Y_{t+j}) = \gamma_{|i-j|}$.

Note : Stationarity of the process (Y_t) is essential. If (Y_t) is not stationary, it is necessary to pretreatment the data to remove trends and periodicity.

3 The non Linear Case

A more general model can be expressed by an equation similar to (1), where the function f is implemented by a multilayer perceptron. Let $W \in \mathbb{R}^m$ denote the vector of parameters $(\mu, \alpha, \beta, \theta)$. So we have

$$\begin{aligned} f(Y_{t-1}, Y_{t-2}, \ldots, Y_{t-p}, W) &= f_W(Y_{t-1}, Y_{t-2}, \ldots, Y_{t-p}) \\ &= \mu + \sum_{j=1}^{k} \beta_j \; \phi\Big(\sum_{i=1}^{p} \alpha_{ji}\, y_{t-i} + \theta_i\Big) \end{aligned} \qquad (3)$$

There are $(p+1)$ input units, k hidden units with a sigmoidal activation function ϕ, and one output linear unit. It is possible to consider exogenous variables, corresponding to additional inputs. The learnt wector $\hat{W} = \text{argmin}(S(W))$ is the least square estimator of W.

ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 157-164

4 The Methodology

We would like to find a model which is as *parsimonious* as possible. Simple models offer robustness, ease of interpretation, and facilitate computations. Many problems are to be solved :

Problem 1 : What is the most convenient method for learning, i.e. for minimizing the error function ?
Problem 2 : How to initialize the weights and the architecture ?
Problem 3 : How to compare two models, how to measure the model quality ?
Problem 4 : How to eliminate the non significant weights so as to decrease the number of parameters ?

1) - First, we choose a second order method, (like BFGS or Levenberg-Marquardt methods),([Luen], or in a neural context [Fomb]) for their reliability and their rapidity. A significant advantage of the method is that it produces the *inverse of the Hessian* $\nabla^2 S$ of the error function $S(W)$ at the minimum $\hat{W}$, as a sub-product.

2) - To initialize the network, we start with one input unit, one hidden unit and all zero weights, except for the μ parameter which is set equal to $\bar{Y}$. Then we progressively add units to the input and hidden layers, computing at each step the modified estimator of the variance. The process is continued as long as this estimator decreases significantly .

3) - The quality of a neural model is typically evaluated by measuring for the performance of the network when it is tested on data (the test set). This can be a good performance evaluation method, but if and only if the test set is representative of the learning set.

We propose to use as in time series one or other Akaike information criterion statistics, the AIC or BIC criterions :

$$\text{AIC} = Log\,\frac{S(W)}{T} + \frac{2m}{T} \quad \text{or} \quad \text{BIC} = Log\,\frac{S(W)}{T} + \frac{m\ \text{Log}\ T}{T}$$

Both criterions contain a penalty term, which gives a cost to any supplementary parameter.

4) - The *more interesting aspect* of the statistical point of view adopted here is that it is possible to know the asymptotic distribution of the least squared estimator of W.

Very recent results by Doukhan and Tsybakov ([Douk]) permit one *to deduce that the least squared estimators are asymptotically Gaussian* (when $T \to \infty$) as in the linear case, due to the boundness property of the function ϕ.

ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 157-164

In fact, when $T \to \infty$, one has

$$\sqrt{T}(\widehat{W} - W) \xrightarrow{\mathcal{D}} \mathcal{N}_m(0, \sigma^2 \Sigma^{-1})$$

where σ^2 is the residual variance, estimated by $S(\widehat{W})/T$, and Σ the Hessian matrix of the error function $S(W)$, multiplied by $(-1/2T)$, and estimated by $(-1/2T)\nabla^2 S(\widehat{W})$. So for each weight parameter w_{ij}, it is possible to build an approximate $(1-\alpha)$ - confidence interval.

4 Weight Elimination

From these results, we propose a new learning algorithm to eliminate the non significant weights. This algorithm is not sensitive to the order of magnitude of the input data, as the Weigend et al. [Weig] algorithm, which eliminates the "small" weights, though does not offer a quantitative measure of "smallness".

The algorithm is a *backward stepwise method* :

1) - Learning is initialized with an appropriate simple architecture.

2) - All the quotients $\widehat{w}_{ij}/\widehat{\sigma}(\widehat{w_{ij}})$ are computed (test statistic of "w_{ij}" $= 0$, against "$w_{ij} \neq 0$")

3) - Weight elimination corresponding to the minimum value of this quotient is attempted, as long as it is smaller than a typical value, say 2. Starting from the weights of the network $\mathcal{N}$, the learning is repeated, and leads to a new network $\mathcal{N}'$.

4) - If $\mathcal{N}'$ is poorer than $\mathcal{N}$ according to the BIC criterion, then $\mathcal{N}$ is kept, otherwise, step 2) is repeated.

Note : Since the approximate variance of the vector $\widehat{W}$ is known, it is possible to test several weights simultaneously.

5 Examples of the Weight Elimination Method

5.1 Application to the Sunspots Activity Data

The SUNSPOTS series gives the annual activity of sunspots since from 1700. These nonstationary series are classically used to compare and evaluate the statistical modelling and forecasting methods. We take the dataset 1700-1920 as training set. The 1921-1935 data can be used as a test set. We compare two classical models and five neural models. See also [Groo)], or [Ginz].

ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 157-164

Model	Number of parameters	Residual variance	BIC
(A)	4	206	5.43
(B)	9	195	5.50
(C)	31	112	5.47
(D)	25	123	5.43
(E)	19	125	5.30
(F)	13	170	5.46
(G)	13	135	5.24

The classical models (A) and (B) that we use are among the best ARIMA models (let B be the lag operator) :

(A) : *Model AR(1,2,9)* : $I - \alpha_2 B - \alpha_2 B^2 - \alpha_9 B^9)X_t = \mu + \epsilon_t$

(B) : *Model ARIMA(1,2,3,4,8,9)(11)(11)* :

$$(I - B^{11})(I - \alpha_1 B - \alpha_2 B^2 - \alpha_3 B^3 - \alpha_4 B^4 - \alpha_8 B^8 + \alpha_9 B^9)X_t = \mu + (I - \beta B^{11})\epsilon_t$$

For the first four neural models, the input vector is only $(X_{t-1}, X_{t-2}, X_{t-3}, X_{t-4})$ and the number of hidden units is decreasing from 5 to 2. The notation *RN(n,k,s)* is for a neural network with n input units, k hidden units and s output units. They are : (C) : *Model RN(4,5,1)*, (D) : *Model RN(4,4,1)*, (E) : *Model RN(4,3,1)*, (F) : *Model RN(4,2,1)*.

For the last neural model, the input vector is $(X_{t-1}, X_{t-2}, X_{t-9}, X_{t-11})$ (the inputs are chosen according to the lags used in the best ARIMA model), with two hidden units : (G) : *Model RN((1,2,9,11),2,1)*.

For the SUNSPOT series, *the neural model is the best, as well for the residual variance, (model (D)), than for the BIC criterion (model (G))*. For the results on the test set, the model (A) gives a residual variance 214, instead of 129 for the neural model (D), for example.

5.2 Progressive Weight Elimination

We apply the weight elimination algorithm of Section **4**, to the model (D) with 4 hidden units. All the connections leading to the 4^{th} neuron of the

hidden layer are not significant. The analysis of the activity of the three hidden neurons in model (E) shows that the non linear parts of the sigmoidal functions are used, and that each neuron makes a distinct contribution to the performance of the network.

5.3 Simulated Examples

We have built artificial series to systematically study the weight elimination method. For a given architecture W, and its associated function f_W, we set $Y_t = f_W(Y_{t-1},\ldots) + \epsilon_t$, where $Y_{t-1},\ldots$ are some previous values of Y, k is the number of hidden units according to the definition of the network, and ϵ_t is a white noise with σ^2 variance Then, we consider a network with more input and hidden units than were used in the network that was used to simulate the data. We trained the network by applying the weight elimination method. In each case, the method leads to a network whose architecture is the same as the artificial one. The estimated weights $\widehat{W}$ are close to the weights W, and the estimated residual variance is a very accurate estimation of σ^2 .

6 Conclusion

The proposed method works very well on relatively small examples. We are now applying it to the EDF series of the daily electricity consumption in France.

Other theoretical questions are of interest. First, we are studying the transposition of the ARIMA models to neural networks with additional input units used to feedback selected *error terms.* In this case, backpropagation is not mathematically justified. Other minimization methods are required and for these one must derive the asymptotic distribution of the estimators, as in the case of feed-forward networks. We are also trying to characterize the non linear series which can be generated by such architectures. Finally, the n-step forecasting problem is difficult to treat with a neural network, due to the possible existence of several fixed points for the recurrence equation which defines the model, some of which are attractive, while others are repulsive. In this matter, we do the same observations than Yao and Tong [Yao] do in another context.

REFERENCES

[Box] BOX G.E.P.,JENKINS G.M., *Time Series Analysis, Forecasting and Control,*Holden-Day, San Francisco (1970).

[Cott] COTTRELL M., GIRARD B., GIRARD Y., Réseaux de neurones et Séries Temporelles, *XXIV Journées de Statistiques,* ASU, Bruxelles (1992).

[Cybe] CYBENKO G., Approximation by Superposition of a Sigmoïdal Function, *Math. Control Signal Systems*, Vol. 2 (1989).

[Dou] DOUKHAN P., TSYBAKOV A., Non-linear ARMA-models : Probabilistic properties and consistent recursive estimation, *Preprint Université Paris-Sud, URA 743 (*1992).

[Fomb] FOMBELLIDA M., DESTINE J., Méthodes heuristiques et méthodes d'optimisation non contraintes pour l'apprentissage des perceptrons multicouches, *Actes Neuro-Nîmes* - 1992.

[Ginz] GINZBERG I., HORN D., Learning the rule of a time series, *Int.J. of Neural Systems,* Vol. 3, N2, pp 167-177 (1992).

[Groo] GROOT C., WURTZ D., Analysis of Univariate Time Series with Connectionnist Nets : A Case Study of Two Classical Examples, *Proceedings of the Munotec Workshop* - Dublin, Dec 1990 and Neurocomputing (in press).

[Horn^a] HORNIK K., STINCHCOMBE M., WHITE H., Multilayer Feedforward Networks are Universal Approximators, *Neural Networks*, 2, pp. 359-366 (1989).

[Horn^b] HORNIK K., Approximation capabilities of multilayer feedforward networks,*Neural Networks*, 4, (1991).

[Lape] LAPEDES A.S., FARBER R., Nonlinear signal processing using neural networks : Prediction and system modeling, *Los Alamos National Laboratory Technical report*, (1987).

[Luen] LUENBERGER D., *Intr. to linear and nonlinear programming*, Addison-Wesley (1973).

[Varf] VARFIS A., VERSINO C., Univariate Economic Time Series Forecasting by Connectionist Methods, *INNC* - Paris (1990).

[Weig] WEIGEND A.S., HUBERMAN B.A., RUMELHART D.E., Predicting the Future : a Connectionist Approach, *Int. J. Neural Systems*, Vol.1, n°3 (1990).

[Yao] YAO Q., TONG H., Quantifying the influence of initial values on nonlinear prediction, *Technical report #UKC/IMS/S92/5c*, University of Kent, U.K. (1992).