ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 351-356

Neural Networks for financial forecast

G. Rotundo*B. Tirozzi‡M.Valente

Abstract.
We use Neural Networks algorithms for forecasting financial time series. We check first the kind of correlations that the series exhibits by means of the estimate of the Hurst' s H parameter. The range of correlations characterize the time series and gives useful hints for choosing the network and the training set. The time series considered is given by the values of the futures of italian BTP.

1. Introduction

We deal with the problem of using neural networks for prediction of financial market time series. The first problem is to have some criteria for choosing the training set and the testing set: there is in fact a great arbitrariness in this choice when dealing with random quantities. We simply select these sets using the condition that they have to be realizations of the same stochastic process. Since we deal with prices of financial market time series we know already that they are realizations of Brownian motion. But the process generating them may be a *fractional Brownian motion* or a simple Brownian motion. The difference between these two processes is crucial for the market theory. In fact the Black and Scholes model cannot be applied if the data are generated by a fractional Brownian motion because Ito calculus fails. In this case the only method for prediction are the neural algorithms. The situation can be even more subtle because the Brownian motion can change its nature during time and so one can get a set of data which correspond for example to a simple Brownian motion and another one which corresponds to a fractional Brownian motion. So we select the data for training and testing in such a way that they are generated by the same kind of Brownian motion. The Brownian motion can be simple or fractional according to the value of the Hurst parameter which is defined in the following section. Applying some suitable criteria for determining the value of H for the daily Italian treasure bond values we get a unique value of H near to one. This means that the daily data are strongly correlated and generated by a fractional Brownian motion. Thus Black and Scholes model cannot be applied for computing the evolution of the prices of the bonds in this case and, as we said above, we have a good reason for applying neural

*Mathematical Department for Economical, Financial and Insurance Decisions, University of Rome "La Sapienza" , Rome, Italy
‡Department of Physics, University of Rome "La Sapienza", Rome, Italy

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 351-356

algorithms. Moreover the existence of the same type of strong correlations, as a consequence of a unique value of H, leads to the use of neural networks also because a strong correlation implies some "rule" generating the data which can then be "learned" by the network. In other words data generated by a simple Brownian motion have independent increments and cannot be "captured" by neural algorithms. In Section 2 we define H and explain the method used for detecting its value. In Section 3 we describe the neural algorithms used and in Section 4 we discuss the special input data constructed starting from the simple time series of the price using current methods for dealing financial or economical values.

2. Estimate of H

Before applying neural networks to our economical time series we make a preliminary analysis on the data estimating their dependence. The neural network can learn a rule from a data set only if there is some law behind them, in other words a learning algorithm will never converge on a set of independent data and so the training set must be chosen inside the range of correlation among data. A quantitative parameter which indicates the degree of dependence among the increments of the process is the Hurst's exponent. It has been introduced by Hurst in 1951 in relation to an hydrology problem of the Nilum basin. It can be defined by means of the variance of the increments:
$E(x_{t+\tau} - x_t)^2 \simeq c\tau^{2H}$.

If [4] 0 <H< 0.5 the series is antipersistent or ergodic, if $H = 0.50$ there is no correlation between the data and if $0.50 < H < 1$ then a long term correlation exists and the more H is close to 1 the more the underlying process has a strong long term component.

We use two different methods for estimating H. The first is the R/S statistics [4] , where R/S means *rescaled range over standard deviation.* Given a random data set $\{X_1, \cdots, X_n\}$ the estimate of H is given

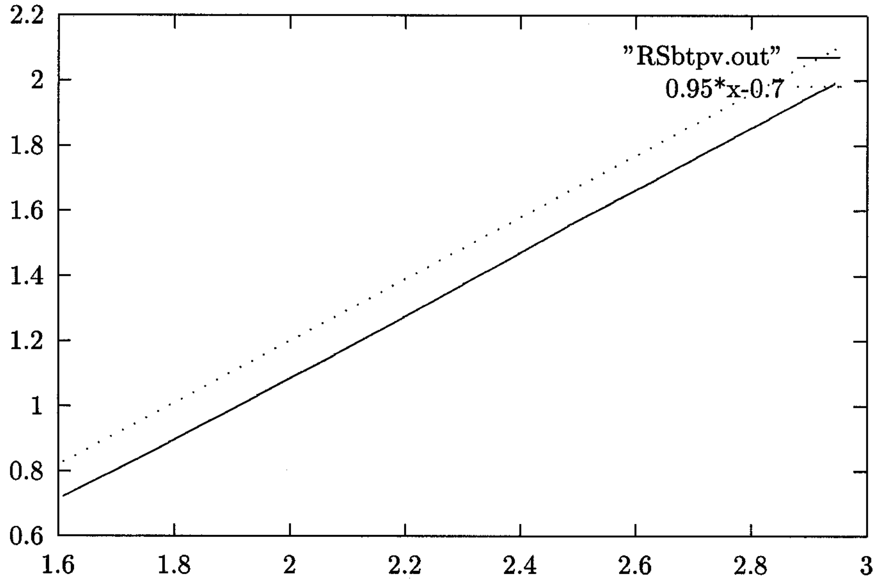
$$E[R/S] \sim an^H \tag{1}$$

for $n \to \infty$ where a is a constant and E is the expectation of the R/S variable with respect to the probability measure of the process generating the time series. The mean is computed averaging among block of data according to the law of large number. H is the slope of the line which best approximates the $(\log n, logE[R/S])$ diagram which can be constructed from the time series ([4]). We got $H = 0.9$ for the BTP futures for all the data that we consider. So we can think that the time series is generated by a unique fractional Brownian motion with very strong correlations. As we mention in the introduction the Black and Scholes model for computing the evolution of prices cannot be used and we have a consistent set of data to which apply neural algorithm.

If the data are scattered too much such a line has no precise meaning and then one can use a spectral method approach. This is not our case but a check

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 351-356

with another independent method is always good because there is also some arbitrariness in the way the statistic R/S is performed.



R/S analysis of the daily closure BTP values not normalized. The dark line represents the results of the R/S analysis, the dashed line is the approximating line 0.95*n+0.7 .

In order to estimate H with spectral methods we use the Fractional ARIMA (FARIMA) models applied to our time series $\{X_1, \cdots, X_n\}$ and a nice theorem found by Hosking [3] in which H is shown to be equal to $d + 1/2$. d is the real power of difference operator $\nabla X_t = X_t - X_{t-1}$ which enters in the definition of FARIMA model: $\nabla^d X_t = \epsilon_t$ where ϵ_t is a process of independent gaussian $N(0, 1)$ variables.

The main theorem for the estimate of H for such processes is given in Geweke [2] by the following :

Theorem

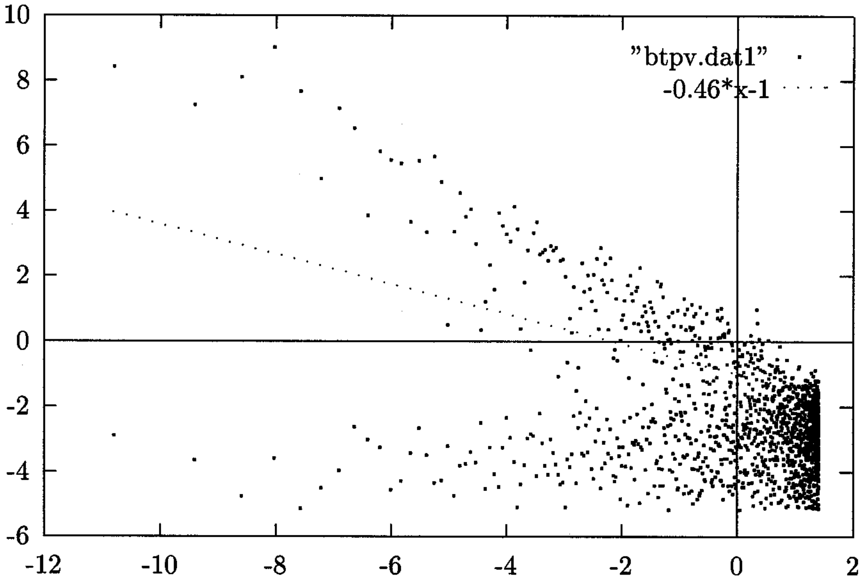
Let $\{X_t\}$ be a general integrated linear process, with $d < 0$. Let $I(\lambda_{j,T})$ be the periodogram of $\{X_t\}$ evaluated at the harmonic frequencies $\lambda_{j,T} = \frac{\pi j}{T}$, where T is the number of sample data. Let $\hat{b}$ be the ordinary estimator of the least square method of b in the regression equation:

$$ln(I(\lambda j, T) = a + b\, ln\left(4sin^2 \frac{\lambda_{j,T}}{2}\right) + \epsilon_{j,T}, j = 1, \cdots, n \qquad (2)$$

Then there exists a function $g(T)$ (such that $lim_{T\to\infty} g(T) = \infty$ and $lim_{T\to\infty} \frac{g(T)}{T} = 0$) such that if $n = g(T)$ then $lim\, \hat{b} = -d$. If $lim_{T\to\infty} \frac{ln(T)^2}{g(T)} = 0$, then $\frac{\hat{b}+d}{\sqrt{(var \hat{b})}} \to N(0, 1)$ where $var\, \hat{b}$ is the usual least square estimator of $var(b)$.

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 351-356

Applying this theorem to the futures of BTP we got the same result of R/S statistics.



Estimate of d for the daily closure values of the BTP. The slope of the dotted line is the estimate of $-d$.

3. Two layer neural networks and genetic algorithms

Since we got strongly dependent random data we feel more sure about the application of neural algorithms to them and since H is uniquely determined we can take rather arbitrarily our training set.

We choose a perceptron with one intermediate layer for dealing with our time series x_t.

We at first construct the input-output pairs $(\mathbf{x}_t, y_t), t = 1, T$, divide them in two subsets of equal magnitude and we perform the learning on the first half of them. Then we perform prediction and training with a moving window procedure in order to minimize the arbitrariness due to the choice of the training set. First we make a prediction on the first pattern that doesn't belong to the training set, $y_{T/2+1}$, and after the couple $(\mathbf{x}_{T/2+1}, y_{T/2+1})$ is taken into the training set. The original neural network is doubled into 2 neural networks that are trained again: the "long range" unit is trained on a training set larger than the one of the "short range" unit. The choice of using two modules of neural nets is connected with the kind of predictions that we want to make: a large set of inputs is of course connected to a prediction of longer range.

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 351-356

The prediction of this 2-modules neural network is simply given the average predictions of each unit. This procedure continues until the last data is reached.

Monte-Carlo training algorithm has been modified in order to have a better performance: if the error relative to a certain pattern does not decrease after a given number of minimizing steps, *ncomp*, then we use the compensation method [6] that adds one or two neurons in order to reduce to 0 the error on that pattern. A measure of the goodness of the forecast, called *fitness* of such nets is connected to the error made on the predictions: $fitness = 1 - \frac{1}{N}\sqrt{\sum_{t=T/2+1}^{T}(y_t - F(\mathbf{x}_t))^2}$ where $(\mathbf{x}_t, y_t)$, $t = T/2+1, T$ are the input-output pairs, $F(x)$ being the output of the neural net under the input $\mathbf{x}$. The genetic algorithms ([5], [1])have been applied on the dimensions of the moving windows in order to maximize the fitness.

4. Application to BTP time series data

A future contract is a contract in which we establish to give a certain amount of a good at a fixed time. These contracts are made on the italian bonds BTP (Treasury Bond).

In order to perform the predictions we got the daily closing values x_t of the futures contracts on BTP from 19/09/91 to 2/10/97.

For the construction of the training couples, $(\mathbf{x}_t, y_t)$ we put $y_t = x_{t+1}$ and we map our time series data x_t in a vector of technical indicators. These parameters are chosen by economical arguments: $MA_k(x_t)$ is a moving average quantity which takes into account the local time evolution while $RSI^l(x_t)$ gives an estimate of the increasing behaviour (relative strength index).

$$\mathbf{x}_t = (MA_3(x_t), MA_5(x_t), MA_8(x_t), MA_{13}(x_t), MA_{21}(x_t),$$
$$RSI^3(x_t), RSI^5(x_t), RSI^8(x_t), RSI^{13}(x_t), RSI^{21}(x_t))$$

where

$$MA_k(x_t) = \frac{1}{k}(x_t) + \frac{k-1}{k}MA_k(x_{t-1}), \tag{3}$$

$$RSI^k(x_t) = \frac{\sum_{i=n-k-1, x_i > x_{i-1}}^{t}(\mid x_i - x_{i-1} \mid)}{\sum_{i=t-k-1}^{n}(\mid x_i - x_{i-1} \mid)} \tag{4}$$

In order to perform the training the x_t values were normalized: denoting by *btpmax* the maximum value of the BTP and by *btpmin* the minimum BTP value that appears in out data and by x_t the BTP value at the day t then the wanted output is given by: $\frac{x_t - btpmin}{btpmax - btpmin}$. In particular we have that $btpmin = 87,187$, $btpmax = 132.15$, italian liras.

The probability mutation of the genetic algorithms was fixed equal to 0.15: this means that for each generation only the 15% of its members can be mutated; the crossover probability was fixed equal to 0.8.

Our results lead to an error (mean of the sum of the absolute values of each error on each prediction) of 0.33 liras on the normalized values, that lead to a

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 351-356

true error of 0.5 liras. Our future aim is to try to decrease this error working on the definition of the training set and the input vector and use other data as input as for example the values of future taken every 5 minutes.

References

[1] Bauer, R.J. jr. *Genetic Algorithms and Investment Strategies*, Wiley Finance edition, 1994.

[2] Geweke, J., Porter-Hudak, S.(1983) *the estimation and application of long memory time series models*, J. Time Ser. Anal. 4 221-238 .

[3] Hosking J. R. M, "Fractional differencing", *Biometrika*, **68(1)**, 1981, pp.165-176.

[4] Lo A. W., "Long-term memory in stock market prices", *Econometrica*,

[5] Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs* , Springer, Copyr. 1996.

[6] Xun Liang, Shaowei Xia, Methods of training and constructing multilayer perceptrons with arbitrary pattern set, International Journal of Neural System, Vol. 6 , No. 3 (September 1995), 233-247, World Scientific Publishing Company