International Journal of Business, Human and Social Sciences
ISSN: 2517-9411
Vol:4, No:4, 2010

Application of Feed-Forward Neural Networks Autoregressive Models in Gross Domestic Product Prediction

E. Giovanis

Abstract**—**In this paper we present an autoregressive model with neural networks modeling and standard error backpropagation algorithm training optimization in order to predict the gross domestic product (GDP) growth rate of four countries. Specifically we propose a kind of weighted regression, which can be used for econometric purposes, where the initial inputs are multiplied by the neural networks final optimum weights from input-hidden layer after the training process. The forecasts are compared with those of the ordinary autoregressive model and we conclude that the proposed regression's forecasting results outperform significant those of autoregressive model in the out-of-sample period. The idea behind this approach is to propose a parametric regression with weighted variables in order to test for the statistical significance and the magnitude of the estimated autoregressive coefficients and simultaneously to estimate the forecasts.

Keywords**—**Autoregressive model, Error back-propagation Feed-Forward neural networks,, Gross Domestic Product

I. Introduction

EMPIRICAL analysis in macroeconomics as well as in financial economics is largely based on times series. The existence of unexpected shocks or innovations to the economy plus measurement errors, strongly suggest that economic variables are stochastic. This approach allows the model builder to use statistical inference in constructing and testing equations that characterize relationships between economic variables. A forecast might be judged successful if it is close to the outcome but that judgment may also depend on how close it is measured. Depending upon the degree of forecast uncertainty, forecasts may range from being highly informative to utterly useless for the tasks at hand. A measure of forecast uncertainty provides an assessment of the expected or predicted uncertainty of the forecast errors which helps to qualify the forecasts themselves and to give a picture of the expected range of likely outcomes.

Aryal and Yao-Wu [1] applied a MLP network with 3 hidden layers to forecast the Chinese construction industry and they compare the forecasting performance of the MLP networks with that of ARIMA and they found that the RMSE of the MLP estimation is 49 percent lower than the ARIMA counterpart. Swanson and White [2,3] applied neural networks to forecast nine seasonally adjusted US macroeconomic time series and they found generally neural networks outperform the linear models. Keles *et al.* [4] developed Adaptive Neuro-Fuzzy Inference System for the prediction of domestic debt presenting very good results.

We propose the specific approach because we are trying to formulate a parametric regression, where we are not able to do it with traditional neural network modelling. Furthermore, in econometrics literature and empirical researches, weighted regressions have been used, where the independent variables are transformed by multiplying them with some weights. In this case we consider the weights of the neural network training.

In this paper we compare the forecasting performance of Autoregressive (AR) and Feed-Forward Neural Networks Autoregressive (FFNN-AR) models in the case of Gross domestic Product. The structure of the paper has as follows. In section II we present the methodology of the estimating and forecasting procedure for both Autoregressive and FFNN Autoregressive models. In section III the frequency and the type of data are described. In section IV the estimated and forecasting results are reported, while in the last section the concluding remarks of this study are presented.

II. Methodology

A. Autoregressive (AR) Models

We consider a series $y_1, y_2, \dots, y_n$. An autoregressive model of order p denoted AR(p), states that y_t is the linear function of the previous p values of the series plus an error term:

$$y_t = \phi_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + \varepsilon_t \qquad (1)$$

, where $\varphi_1, \varphi_2 \dots, \varphi_p$ are weights that we have to define or determine, and ε_t denotes the residuals which are normally distributed with zero mean and variance σ^2 [5]. Based on t-statistics, we choose the lag order. We test up to 5 lags.

Eleftherios Giovanis is with the Royal Holloway University of London, department of Economics, Egham, Surrey TW20 0EX, UK, e-mail: giovanis@freemail.gr, Eleftherios.Giovanis.2010@live.rhul.ac.uk

.

Conditioned on the full set of information available up to time *i* and on forecasts of the exogenous variables, the one-period-ahead forecast of y_t would be

$$\hat{y}_{t+1|t} = \hat{\phi}_{0\ t+1|t} + \hat{\phi}_1 y_t + \hat{\phi}_2 y_{t-1} + ... + \hat{\phi}_p y_{t-p+1} + \hat{\varepsilon}_{t+1|t} \quad (2)$$

B. Feed-Forward Neural Networks Autoregressive (FFNN-AR) Models with Error Back-Propagation Algorithm

The Feed-Forward Neural Networks model is a widely used approach known for its speed and accuracy. A FFNN can be represented as in Fig. 1. To be specific, in Fig. 1 we present a feed-forward neural network with an input layer of m_0 nodes for $n=1 .... m_0$, one hidden layer and a single output layer. The input layer includes the input variables, which in the case we examine are the lags of the dependent variable of (1) and specifically the Gross Domestic Product of each country. The hidden layer consists of hidden neurons or units placed in parallel. Each neuron in the hidden layer performs a weighted summation of the weights which then passes an activation function. The output layer of the neural network is formed by another weighted summation of the outputs of the neurons in the hidden layer [6].

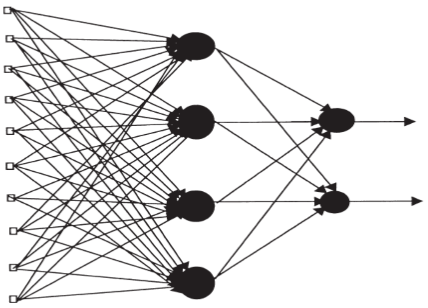

Fig. 1 A Feed-Forward neural networks with one hidden layer and one output layer

The FFNN model is estimated based on the error backpropagation algorithm [6]-[7]. This algorithm adopts a learning process referred as error correction learning. Specifically the learning process has as the main target the minimization of the cost function leading to a learning rule known as the *Delta rule* or *Widrow-Hoff rule* [8]. The cost function which is minimized is defined as:

$$e_k(n) = d_k(n) - y_k(n) \quad (3)$$

, where $e_k(n)$ is the error signal, $y_k(n)$ is the neural network output signal and $d_k(n)$ is the desired target, which is the real value of the Gross domestic Product growth rate. The purpose of the neural network learning process is to apply corrective adjustments to the synaptic weight of neuron *k* in order to make the output $y_k(n)$ to come closer to the desired response $d_k(n)$ in a step-by-step manner. The minimization of the cost function is:

$$f(n) = \frac{1}{2} e_k(n) \quad (4)$$

We denote the $w_{kj}(n)$ as the value of the synaptic weight w_{kj} of neuron *k* excited by element $x_j(n)$ on the signal input vector $x_j(n)$ at time step *n*, where input vector contains the independent variables we examine. Based on the *delta rule* the adjustment $\Delta w_{kj}(n)$ applied to the synaptic weight $w_{kj}(n)$ at time step *n* is given by the following relation:

$$\Delta w_{kj}(n) = \eta e_k(n) x_j(n) \quad (5)$$

,where the Greek letter η denotes the learning rate. After the computation of the synaptic adjustment $\Delta w_{kj}(n)$ the synaptic weight $w_{kj}(n)$ is updated in the following way.

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n) \quad (6)$$

In other words $w_{kj}(n)$ and $w_{kj}(n+1)$ can be viewed as the old and new values respectively [6]-[7]. Additionally in this paper we update the weights with learning rate, as also with momentum rate, so we have:

$$w_{kj}^{*}(n+1) = w_{kj}(n+1) + mom^{*}(w_{kj}(n+1) - w_{kj}(n)) \quad (7)$$

,where the *mom* denotes the momentum rate. The hidden neurons used are equal with the number of inputs. In the case we have an autoregressive model with constant, then the hidden neurons are equal with the number of the dependent lagged series, while in the case we obtain constant too then the number of hidden neurons is equal with the number of the dependent lagged series plus one indicating the constant, which is a vector of ones and specifically is the bias in neural network model. We test three transfer functions, from input to hidden layer, the logistic, hyperbolic tangent and linear. On the other hand the linear transfer function, from hidden to output layer, is used in all three tests. The logistic, hyperbolic tangent and linear transfer functions are defined respectively by expressions (8)-(10).

$$f(x) = \frac{1}{1+e^{-x}} \quad (8)$$

$$f(x) = \frac{2}{1+e^{-2x}} - 1 \quad (9)$$

$$f(x) = x \quad (10)$$

The process from the input to output layer is the forward pass, where the inputs *x* are fed in to the network. The transfer functions at the nodes and their derivatives are evaluated in each node and then derivatives are stored. The purpose of the backpropagation algorithm, which is the backward pass from output to input layer, is the derivation of (11)-(13). This can be written respectively for logistic, hyperbolic tangent and linear transfer functions respectively as:

$$\frac{d}{dx}f(x)=\frac{e^{-x}}{(1+e^{-x})^{2}}=f(x)\cdot(1-f(x)) \qquad (11)$$

$$\frac{d}{dx}f(x)=f'(x)=(1-f(x)^{2}) \qquad (12)$$

$$\frac{d}{dx}f(x)=f'(x)=1 \qquad (13)$$

More specifically the first step is the forward pass. The second step is the backpropagation to the output layer. This can be written as:

$$e^{(B)}{}_{j}=y_{j}(1-y_{j})(y_{j}-d_{j}) \qquad (14)$$

, where $e^{(B)}{}_{j}$ is defined as the backpropagation error, y_j is the signal or output of the output layer and d_j is the desired output, in our case is the actual gross domestic product growth rate. The partial derivative is:

$$\frac{\partial E}{\partial w^{(\mathrm{B})}{}_{kj}}=y_{j}(1-y_{j})(y_{j}-d_{j})\cdot ho_{j}=e^{(B)}{}_{j}\cdot ho_{j} \qquad (15)$$

, where E is the error-cost function (3), ho_j denotes the output values from the hidden layer and $\partial w^{(\mathrm{B})}{}_{kj}$ is the synaptic weight matrix from output to hidden layer. The next step of backpropagation algorithm is the backpropagation to the hidden layer. This is:

$$e^{(A)}{}_{j}=ho_{j}(1-ho_{j})\sum_{k=1}^{n}w_{kj}{}^{(B)}e^{(B)}{}_{j} \qquad (16)$$

, where $e^{(A)}{}_{j}$ is defined as the backpropagation error to hidden layer, ho_j , $e^{(B)}{}_{j}$ and $\partial w^{(\mathrm{B})}{}_{kj}$ are defined as previously. The partial derivative is:

$$\frac{\partial E}{\partial w^{(A)}{}_{kj}}=-e^{(B)}{}_{j}\cdot I_{k} \qquad (17)$$

I_k denotes the inputs and $\partial w^{(A)}{}_{kj}$ is the synaptic weight matrix from hidden to input layer. Finally we propose the following neural network regression:

$$y=\sum_{i=1}^{p}(\phi_{\mathrm{i}}y_{t-i}+\mathrm{b})w^{(A)}{}_{kj} \qquad (18)$$

, where for i=1,2…p is the number of lags as in the case of AR models, $w^{(A)}{}_{kj}$ are the optimized weights from hidden to input layer and b is the bias which is a vector of ones as in the case of the ordinary least squares method. So we regress the initial dependent variable y, which denotes the GDP growth rates, on the weighted inputs, forming a kind of a weighted regression. The forecasting performance of Autoregressive (AR) models and Feed-forward neural networks Autoregressive (FFNN-AR) models in both in-sample and out-of- sample periods is counted based on the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) described respectively by (19) and (20).

$$MAE=\frac{1}{n}\sum_{i=1}^{n}|\,y_{i}-\hat{y}_{i}\,| \qquad (19)$$

$$RMSE=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_{i}-\hat{y}_{i})^{2}} \qquad (20)$$

III. DATA

The data are in quarterly frequency and are referred in Gross Domestic Product (GDP) growth rates for quarter-by-quarter. The period examined is 1991-2009 for France, Italy, UK and USA. Moreover the period 1991-2006 is obtained as the in-sample for AR model or as the train period for the FFNN-AR model, while period 2007-2009 is taken as the out-of-sample period. Moreover, we apply a four step ahead period forecasting. Specifically in the one-step ahead prediction both models present a very similar and high performance. The purpose is to extend the step forecasting period because it is much more useful. Firstly, we estimate the forecasts for 2007, then we replace the forecasting values with the actual and we estimate the forecasts for 2008. The same procedure is followed for 2009.

IV. EMPIRICAL RESULTS

In Table 1 we observe that based on ADF test [9] the GDP growth rate is not stationary in its levels in the case of France for all significance levels, while the GDP growth rate in Italy and USA is marginally stationary in 0.10. For this reason we consider that first differences are more appropriate. Moreover it should be noticed that in both AR and FFNN-AR models we excluded constant or bias because the forecasts as also the estimated coefficients are much more significant in the case we do not take constant. The learning and momentum rates have been set up at 0.05 and 0.1 respectively, the goal error at 0.5 and the number of maximum epochs at 50. The algorithm runs until the network error reach a level lower than the goal error.

In Table II we report the epochs and the networks error reached after the neural network training. It should be noticed that all activation functions from input-to-hidden layer present the same forecasting performance so we choose randomly the logistic function. In all cases we found that the models present higher forecasting power with no constant in the estimated regression. Additionally, based on *t-statistics,* the optimum lag order is 1.

TABLE I
ADF UNIT ROOT TESTS

	France	Italy	UK	USA	Critical values for ADF[1]
ADF test					-4.110 α=0.01
In Levels	-2.039	-3.26	-3.81	-3.17	-3.482 α=0.05
In First differences	-6.25	-6.40	-6.70	-6.45	-3.169 α=0.10

[1] MacKinnon, [9]

TABLE II
EPOCHS AND NETWORK ERROR REACHED
AFTER THE TRAINING PROCESS

	France	***Italy***	***UK***	***USA***
Goal error	0.5	0.5	0.5	0.5
Error after optimization	0.499	0.490	0.499	0.482
Epochs after optimization	15	25	12	15

TABLE III
AUTOREGRESSIVE (AR) ESTIMATIONS

France

$$y_t = \underset{(2.382)^{**}}{0.2826}\, y_{t-1}$$

$R^2_{adj.} = 0.0799$, LBQ^2 (2)=0.0298, p-value for LBQ^2 =0.863
ARCH-LM(2)=0.212, p-value for ARCH-LM=0.9789

Italy

$$y_t = \underset{(3.587)^{*}}{0.4158}\, y_{t-1}$$

$R^2_{adj.} = 0.1649$, LBQ^2 (2)=0.1544, p-value for LBQ^2 =0.694
ARCH-LM(2)=0.0937, p-value for ARCH-LM=0.9106

UK

$$y_t = \underset{(2.469)}{0.2861}\, y_{t-1}$$

$R^2_{adj.} = 0.0632$,LBQ^2 (2)=1.7913, p-value for LBQ^2 =0.181
ARCH-LM(2)=1.502, p-value for ARCH-LM=0.2306

USA

$$y_t = \underset{(11.099)^{*}}{0.2296}\, y_{t-1}$$

$R^2_{adj.} = 0.0479$,LBQ^2 (2)=1.3274, p-value for LBQ^2 =0.249
ARCH-LM(2)=0.842, p-value for ARCH-LM=0.4354

t-statistics in parentheses, * denotes significance in α=0.01, ** denotes significance in α=0.05, LBQ^2 is the Ljung-Box test on squared standardized residuals with 2 lags, ARCH-LM denotes Lagrange Multiplier test for ARCH effects with 2 lags

The estimated Autoregressive (AR) and Feed-Forward Neural Networks Autoregressive (FFNN-AR) results are reported in Tables III and IV respectively. Based on LBQ^2 and ARCH-LM tests we reject the existence of autocorrelation and ARCH effects respectively in residuals [5].

In Tables V and VI we present the MAE and RMSE measures for the in-sample and the out-of-sample periods respectively of the estimated models. Only in one case AR model outperforms the FFNN-AR model in the in-sample period and more specifically in the case of Italy, while in the remained countries we examine, RMSE and MAE are very close among the two models. On the other hand in the out-of-sample period which is of greatest interest, FFNN-AR outperforms significant AR model. This can be shown also from Fig. 2-5. We conclude that with the approach we propose we gain two things. Firstly, we get an alternative autoregressive procedure and secondly, we improve the forecasts n the out-of-sample period.

TABLE IV
FEED-FORWARD NEURAL NETWORKS AUTOREGRESSIVE
(FFNN-AR) ESTIMATIONS

France

$$y_t = \underset{(25.773)^{*}}{0.3720}\, y_{t-1}$$

$R^2_{adj.} = 0.4081$, LBQ^2 (2)=0.0789, p-value for LBQ^2 =0.784,
ARCH-LM(2)=0.9125, p-value for ARCH-LM=0.6073

Italy

$$y_t = \underset{(18.578)^{*}}{0.5014}\, y_{t-1}$$

$R^2_{adj.} = 0.7847$, LBQ^2 (2)=0.8915, p-value for LBQ^2 =0.348,
ARCH-LM(2)=1.5791, p-value for ARCH-LM=0.4540

UK

$$y_t = \underset{(32.573)^{*}}{0.1377}\, y_{t-1}$$

$R^2_{adj.} = 0.6744$, LBQ^2 (2)=1.9817, p-value for LBQ^2 =0.168,
ARCH-LM(2)=1.607, p-value for ARCH-LM=0.2847

USA

$$y_t = \underset{(32.139)^{*}}{0.1110}\, y_{t-1}$$

$R^2_{adj.} = 0.6358$, LBQ^2 (2)=1.9915, p-value for LBQ^2 =0.170,
ARCH-LM(2)=1.097, p-value for ARCH-LM=0.3541

t-statistics in parentheses, * denotes significance in α=0.01, LBQ^2 is the Ljung-Box test on squared standardized residuals with 2 lags, ARCH-LM denotes Lagrange Multiplier test for ARCH effects with 2 lags

TABLE V
FORECASTING PERFORMANCE OF AR AND FFNN-AR
MODELS FOR IN-SAMPLE PERIOD

	AR		***FFNN-AR***	
	MAE	RMSE	MAE	RMSE
France	0.5201	0.6371	0.5040	0.6148
Italy	0.5808	0.7566	0.5841	0.7602
UK	0.4196	0.5341	0.4146	0.5295
USA	0.5481	0.6606	0.5267	0.6383

TABLE VI
FORECASTING PERFORMANCE OF AR AND FFNN-AR
MODELS FOR OUT-OF-SAMPLE PERIOD

	AR		***AR***	
	MAE	MAE	MAE	MAE
France	2.1385	2.5840	1.5501	1.9210
Italy	2.3555	2.9334	1.9911	2.2682
UK	2.2230	2.8528	1.8801	2.3076
USA	2.0581	2.5179	1.8832	2.2711

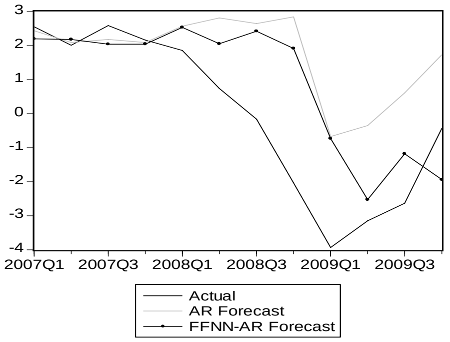

Fig. 2 Out-of-sample period forecasts with AR and FFNN-AR models for France

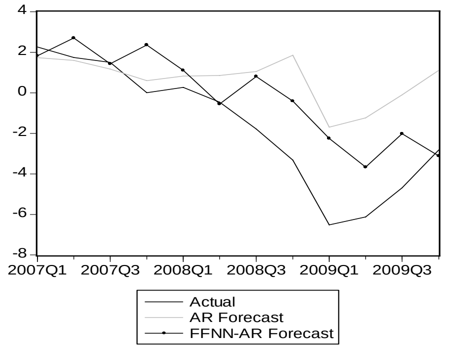

Fig. 3 Out-of-sample period forecasts with AR and FFNN-AR models for Italy

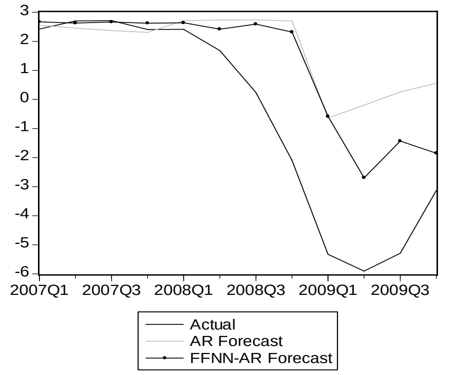

Fig. 4 Out-of-sample period forecasts with AR and FFNN-AR models for UK

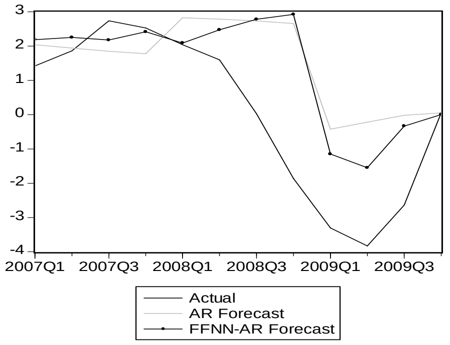

Fig. 5 Out-of-sample period forecasts with AR and FFNN-AR models for USA

V. CONCLUSIONS

The main conclusion of this paper is that the forecasting performance of the model we propose is much more superior in the out-of-sample period, while its forecasting ability is very close with that of AR model in the in sample period. Additionally, genetic algorithms for the training process can be used instead to error backpropagation training or the procedure can be enriched with other nonlinear optimization procedures, like Levenberg-Marquardt among others.

REFERENCES

[1] R.D. Aryal and W. Yao-Wu, "Neural Network Forecasting of the Production Level of Chinese Construction Industry", *Journal of Comparative International Management* , vol. 29, pp. 319-33, 2003

[2] N.R Swanson and H. White, "A model selection approach to real time macroeconomic forecasting using linear models and artificial neural networks", *Review of Economics and Statistics*, vol. 79, pp. 540-650, 1997a

[3] N.R Swanson and H. White, "Forecasting economic time series using adaptive versus non-adaptive and linear versus nonlinear econometric models", *International Journal of Forecasting*, vol. 13, pp. 439-461, 1997b

[4] A. Keles, M. Kolcak and A. Keles, "The adaptive neuro-fuzzy model for forecasting the domestic debt", *Knowledge-Based Systems* , vol. 21, no. 8, pp. 951-957, 2008

[5] W.H Greene, *Econometric Analysis*, Sixth Edition, Prentice Hall, New Jersey, 2008, pp. 560-579

[6] Haykin, S. (1999), *NEURAL NETWORKS: A Comprehensive Foundation*, Second Edition Pearson education, Prentice Hall, Delhi, India, 1999, pp. 33-47, 73-76

[7] Graupe, D. *Principles of Artificial Neural Networks,* Second Edition, Advanced Series on Circuits and Systems, 6, World Scientific Co., Singapore, 2007, pp. 10-15, 20-24, 59-63

[8] B. Widrow, and M. Hoff, E., "Adaptive switching circuits," In *Western Electronic Show and Convention Record, Institute of Radio Engineers (now IEEE)*, vol. 4, pp. 96-104, 1960

[9] D. A. Dickey, and W. A. Fuller, "Distribution of the Estimators for Autoregressive Time Series with a Unit Root", *Journal of the American Statistical Association*, vol. 74, pp. 427–431, 1979

[10] J. G. MacKinnon, "Numerical Distribution Functions for Unit Root and Cointegration Tests", *Journal of Applied Econometrics*, vol. 11, pp. 601-618, 1996