HYBRID NEURAL NETWORKS AS A NEW APPROACH IN TIME SERIES FORECASTING

[a]LUKÁŠ FALÁT

University of Žilina, Faculty of Management Science & Informatics, Univerzitná 8215/1, 010 26 Žilina
email: [a]Lukas.Falat@fri.uniza.sk

Abstract: Nowadays there exists a various number of ways for time series prediction. Econometric analysis includes mainly ARIMA and GARCH models. However, in recent years, as computers become more and more the part of our lives, methods of machine learning have been used; among them artificial neural networks. This work combines these two independent and different prediction models into one combined model for time series prediction as it is an assumption that the combination could achieve better prediction results than the individual models. This new approach is then applied to exchange rates data to illustrate the sense of using this hybrid modelling technique.

Keywords: time series, ARIMA, neural networks, RBF, hybrid modeling, AUD/USD.

1 Introduction

The predictions play very important role in various areas of people's lives – demographic predictions, industrial planning, geographical expectations, water consumptions and so on. Also, financial analyzers try to predict the future value of commodities, stocks or exchange rates. Resulting from this, nowadays, more than ever, making precise predictions is a must. To achieve these kind of predictions, various approaches are applied. The most used approach, which has been used for many years, is a statistical approach. This approach is represented by ARIMA, GARCH, Exponential Smoothing, Kalman filter, linear regression and so on. However, it has been showed that this technique does not always provide sufficient results. It can be caused by a complexity of real problems or because of any other reason.

Therefore, other methods, using mainly the power of computers, have been created. Among these methods of machine learning, artificial neural networks, inspiriting by a human neural network, have become very popular. Today, these ANN models have become the interest of many prediction analyzers.
The idea of better predictions results from making some combination of these two model in order to achieve very accurate future values of any variable. This idea is not at all new. The hybrid models have been created and then applied and used in many areas. The purpose is to combine some good qualities from individual models to better the results in whatever subject.
As for time series predictions, researchers have proposed (among others) to hybridize ARIMA models together with ANN models to create one hybrid models presuming to have better predictive results. This presumption will therefore be tested in this work in order to find out whether it is worth making this type of models.
The hybrid model will be tested on economic time series, more specifically on exchange rate of AUD/USD. Tested data include 1044 observations, from 03/01/2007 to 03/01/2011.

What for other section of this work; in section 2, the ARIMA statistical models, which will be used to create hybrid model, will be discussed. Section 3 will deal with artificial neural networks. Section 4 will talk something more about hybrid models. In section 5, the particular tested hybrid model will be presented and its forecast results will be compared to individual models. Section 6 concludes this work.

2 ARIMA Models

ARMA models, also known as Box-Jenkins models are an excellent tool in statistical time series modelling if the current value of variable linearly depends on the previous values of the same variable or if the current value of random part depends on the previous values of random part. Formally, ARMA model can be expressed as

$$y_t = \sum_{i=1}^{p} \phi_i y_{t-1} + \sum_{j=0}^{q} \theta_j \varepsilon_{t-1} \tag{1}$$

This model is composed of two parts. The first, autoregressive part, represented by autoregressive parameters $(\phi_1, \phi_2, ...)$ is deterministic; and the second part, also known as the moving average part represented by independent random parts $(\varepsilon_t, \varepsilon_{t-1}, \varepsilon_{t-2}, ...)$ is stochastic. The number of *p* determines AR order, the number of *q* determines MA order in ARMA(p,q) model.

In order one can model time series using ARIMA models, the time series has to be stationary, or at least is has to be weak-stationary. It means the time series has to be statistically unchangeable in its expected value and variance (first and second moment). But in real world, there exists a huge number of problems where the observing time series is not stationary in. Therefore ARIMA models, as an extension to ARMA models, can be applied. Let y_t be a time series. y_t will be called ARIMA(p,d,q) process if its d^{th} differences produce ARMA(p,q) process. ARIMA can be formally defined as

$$\Phi(B)(1-B)^d y_t = \mu + \Theta(B)\varepsilon_t \tag{2}$$

It is also obvious that if d equals zero, ARIMA equals just simple ARMA process.

Another option, except of using ARIMA models, how transform non-stationary time series into stationary includes (if possible) for example detrending or logarithmic calculation.

Process of making a statistical model have more steps. The first step, after data collection and analysis, is identification of model. Identification is usually done using graphical representations of autocorrelation and partial autocorrelation functions, known as ACF and PACF. There should be at least K <= N/4 values in the ACF and PACF graph. $\pm 2/\sqrt{N}$ is a border between statistically significant and non-significant value in the graph. The table 1 introduces the basic rules for model identification.

	ACF	PACF
MA(q)	Cut off after lag q	Exponential or sinusoid decline
AR(p)	Exponential or sinusoid decline	Cut off after lag p
ARMA(p,q)	Exponential or sinusoid decline	Exponential or sinusoid decline

Table 1 Theoretical ACF and PACF of Box-Jenkins models

Apart from ACF and PACF, model identification can be also performed via other tools, for example various information criterions (AIC, SIC, BIC).

After model identification, quantification of model using a statistical computer program (R, Eviews, Matlab, SPSS) is performed. Diagnosis of model (and its residuals) is then performed in order to find out whether our model correctly model the specified time series. If the model is not correct, it had to be repaired or another part has to be incorporated into the model (f. ex. GARCH). If so, quantified model is evaluated and then predictions are made. The evaluation characteristics for quantified model as well as future prediction include among others MSE (Mean Square Error), RMSE (Root of MSE) or MAPE (Mean Absolute Percentage Error).

$$MSE = \frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2 = \frac{1}{N}\sum_{t=1}^{N} e_t^2 \tag{3}$$

$$MAPE = \frac{1}{N}\sum_{t=1}^{N}\frac{|y_t - \hat{y}_t|}{y_t}100 = \frac{1}{N}\sum_{t=1}^{N}\frac{|e_t|}{y_t}100 \tag{4}$$

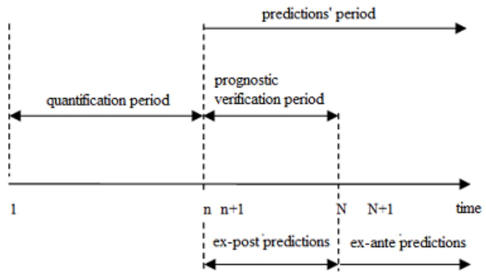

Figure 1 Time axis division in time series modelling

3 Artificial Neural Networks

As having said before, artificial neural networks representing a part of machine learning techniques can be used to making valuable time series predictions. Since appropriate to have some feedback for network learning in this case, supervised ANNs will be discussed. The goal of ANN is to find an input-output function so that the output would have desired parameters and a predicted error would be minimal.

Let F: $x_t \in R^k \to y_t \in R^1$ be a projection assigning k-dimensional vector of inputs $x_t^T = (x_{1t}, x_{2t}, \ldots, x_{kt})$ one dimensional output y_t in specific time moment t. Let G: $G(x_t, w_t): x_t \in R^k_{train} \to y_t \in R^1_{train}$ be a restriction of F. The task is then to find the values of w_t so that functional values of G would be so close to desired output as it is possible. Let E(w) be a function

$$E(w_t) = \sum_{x_t, y_t \in R^k_{train}} (G(x_t, w_t) - y_t) \tag{5}$$

This function will represent squares of deviations of function G from expecting values of function F. If a minimum is found, G is adapted for approximation of F.

The simplest model of the mathematical model of neuron discussed above is perceptron. It is a feed-forward type of network, so it contains only forward relations realised only from lower layers to higher. Architecture of feed-forward neural network can be seen on figure 2.

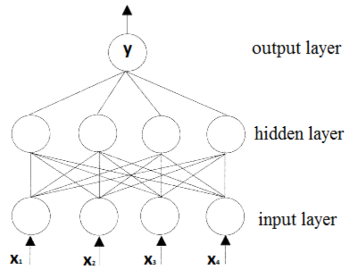

Figure 2 Feed-forward Artificial Neural Network with 4 Inputs and 1 Output

As seen in the figure 2, input set is represented by the vector $x_t^T = (x_{1t}, x_{2t}, \ldots, x_{nt})$ and output layer containing usually only one neuron is represented by the network output. In most cases there is also hidden layer containing hidden neurons. Using $G(x_t,w_t)$, the inputs are transformed into output when going through the network. The produced output is then compared to the historical output of a series.

Learning or so called training of the network means adaption of weights between inputs and hidden neurons and between hidden neurons and the output. The aim of learning is to have a trained network so that G would approach F the most. Then one can say that ANN became an expert in specific area of data which had been trained on. Learning is performed on training set data (see fig 1).

The learning of feed forwards ANNs (perceptron, RBF) is based on back propagation algorithm. ANN counts its output on the base of its inputs by counting potentials and then outputs (activated potentials) of hidden neurons and an error of the whole network for specific input is counted. Afterwards an error is back propagated into the network and the weights are adapted on the base of this error. This iterative procedure is performed (for every input of training set) while the network is considered to be adapted. It is the moment when error function achieves a minimum.

Once the networks has been trained, the network is then validated. Validation is performed on validation set (see fig 1) and consists in making ex-post predictions. The evaluation of ex-post predictions, as well as networks training and ex-ante predictions, is usually checked by characteristics like RMSE or MSE. When ex-post predictions evaluation is made, we can proceed to make ex-ante predictions by this ANN.

Due to some cons of perceptron network, the more sophisticated version of feed forward neural network has been created. This network is called RBF and the architecture is quite similar to perceptron, however there are some differences. RBF has its name due to radial basic activation functions (RBF) in activation of hidden neurons which is different from sigma function of perceptron. In addition, calculation of potentials of hidden neurons is calculated as Euclidean distance given by vectors $u^j = \| x - w^j \|^2$ and not by just scalar product of *X* and *W* as at perceptron. Thanks to these modifications, RBF provides much better predictions results as perceptron.

4 Hybrid Neural Networks

Generally, hybrid model is any combination of two or more independent models. These models are integrated into one complex producing only one output in specific time t. The purpose of this operation is to raise the prediction accuracy of the model.

Nowadays, there exists several types of hybrid models, such as fuzzy networks plus ANN, ARIMA plus ANN or ANNs with other specialized systems. As mentioned in section 1, this work will deal with ARIMA plus ANN hybrid models.

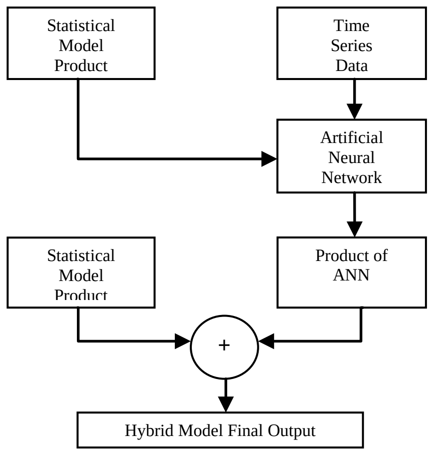

Figure 3 General ARIMA plus ANN hybrid model schema

As the reason for constructing hybrid model is to have better forecasts, the main point here is to find out how to combine independent models in order to produce the best possible results. This is done by correctly constructed hybrid schemas. Figure 3 illustrates basic ARIMA plus ANN hybrid model.

Seeing on figure 3, the inputs of ANN are not only time series data (just like at classical ANN) but also a product of constructed statistical model enters this ANN. This product of statistical model can include residuals, outputs estimations or predictions. This ANN then produces the output on which a linear technique could also be applied to produce final combined hybrid model output (however it is not necessary).

Since this is relatively new and experimental approach in time series modeling, very dependent on chosen hybrid model schema, it is no surprise that the increase of predictions is possible, however not guaranteed. Also for that reason, it is good to perform comparison with classical modelling approaches.

5 Tested Hybrid Model and Forecasting Results

Theoretical aspects of hybrid modelling described above will be applied to real time series data. Data used to illustrate the sense of using ARIMA plus ANN hybrid model were exchange rates of AUD/USD. These are daily data taken from 03/01/2007 to 03/01/2011. The number of data is 1044 and the data were downloaded from the following site: http://www.global-view.com/forex-trading-tools/forex-history.

Because of test reasons, the observed data were divided into two parts – the training set contained 1002 values (03/01/2007 – 12/31/2010) and the validation set (for model verification) contained 42 observed values (01/01/2011– 03/01/2011). The ANN modeling was performed by self-constructed application and econometric models were quantified in Eviews.

For hybrid model testing, the following hybrid schema illustrated on figure 4 was chosen.

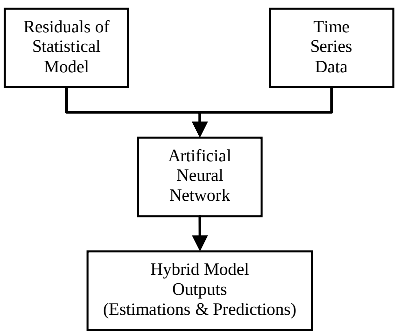

Figure 4 Schema of Tested Hybrid Model

The assumption was that this network could better the predictions' results of individuals models. For network testing, RMSE evaluative characteristics were used. Except for perceptron and classical RBF network, the extensions of RBF (soft RBF, cloud RBF, granular RBF) which have tendency of better estimations. For details of these RBF extensions, see [3].

Modelling time series using ANN application demands performing plenty of experiments (and of course a lot of time) since ANN contains huge number of parameters (speed of learning, number of hidden neurons, input neurons, number of epochs, size of training set, size of validation set,…) to set up. Finally, the best results which were achieved are in the following table 2.

	Hybrid model (perceptron)	**Hybrid model (RBF)**
Type of ANN		Cloud RBF
Number of hidden neurons	5	8
Speed of Learning	0,10	0,10
Number of epochs	2000	2000
Training set (A)	1001	1001
Validation set (E)	42	42
$RMSE_A$	0,0001110	0,0074812
$RMSE_E$	0,0055365	0,0032273

Table 2 Best Achieved Approximate and Predictive Results evaluated by RMSE

To show the effectiveness or ineffectiveness of this constructed hybrid model, comparative analysis with individual model were performed.

What for econometric model, AUD/USD seemed to be AR(0) process with only a constant. However, it later showed that residuals did not create a white noise process and there are some dependencies in residuals. Therefore the model had to be repaired. And as exchange rates are high frequented data, they are very dynamic and a volatility is changing and is not constant over time, there was an presumption of ARCH effect. Because of this, (G)ARCH models were implemented in the model. In this specific case, the extension of classical (G)ARCH – TGARCH provided the best results. After that, diagnosis of the model was performed again and the model was confirmed to be OK. The residuals from this model were than the inputs into the ANN network. For (G)ARCH models details can be found in [1] [5].

The numerical comparison of all tested and quantified models are shown in table 3. The models are ordered according to root mean squared error achieved for new data (ex-post predictions) from validation set. Training and validation set was always set to 1001/42 and number of epochs was set to 2000 for every model.

	Type of Model	**Hidden Neurons**	**$RMSE_E$**	**$RMSE_A$**
1.	Hybrid (Cloud RBF)	8	0,003227	0,007481
2.	Granular RBF	7	0,005316	0,031223
3.	Hybrid (Perceptron)	5	0,005536	0,000111
4.	AR(0) + TGARCH(1,1,1)	--	0,006205	0,009300
5.	RBF	12	0,009063	0,023915
6.	Perceptron	4	0,046615	0,009573

Table 3 Approximate and Predictive Accuracy of Various Types of Models of AUD/USD (Measured by RMSE)

As we can see from table 3, the assumption that RBF network provides better results than perceptron network has been confirmed too. In addition, extensions of RBF (cloud, granular RBF) also provided better results than classical RBF network.

6 Conclusion

The aim of this paper was to apply a new approach in time series modelling – hybrid models. After theoretical aspects of ARIMA models, ANNs and hybrid models, one specific hybrid model scheme was constructed. After the construction, the model was thoroughly tested to find out whether it is worth building hybrid models.

The tested hybrid model proved excellent results and its numerical characteristics overcame individual models (ANN, statistical models) in this case. Actually, according to $RMSE_E$ errors, it was the best model of all tested models. What is also very surprising is the fact that also perceptron overcame a statistical model. The tested hybrid model provided the best results with cloud RBF network.

However, even if hybrid modelling can cause an improvement of statistical or neural network model, it is certainly not an always-rule. One of the most significant factors on the improvement or unimprovement of the prediction results is hybrid model schema which is used.

A slight disadvantage of hybrid approach in time series modelling is a fact that one has to have products of ANN and products of statistical model as well. So it is necessary to create two independent models (one statistical model and one ANN model with data inputs and statistical product inputs as well). It is therefore obvious that hybrid approach demands more time for time series modelling than an individual approach. In addition to this, this hybridization does not have to lead to better predictions. However, in today's world full of complex problems with linear and non-linear relations, hybrid model can be more effective solution to specific problem than individual models and because of this hybrid modelling has definitely the sense and it is certainly worth trying modelling real problems in this way.

Literature:

1. ENGLE, R.F.: *Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of United Kingdom Inflation*, Econometrica, Vol. 50, No. 4, 1982. p987 – p1008.
2. LEK, S., GUÉGAN, J.F.: *Artificial neural networks as a tool in ecological modeling*, Ecological Modelling 120, 1999. p65–p73.
3. MARČEK, D., MARČEK, M.: *Neurónové siete a ich aplikácie*, Žilina: EDIS – Vydavateľstvo ŽU, 2006. 223p. ISBN 80-8070-497-X.
4. MARČEK, D.: *Some Intelligent Approaches to Stock Price Modelling and Forecasting*, Journal of Information, Control and Management Systems, Vol. 2, 2004.
5. MARČEK, M.: *Viacnásobná štatistická analýza dát a modely časových radov v ekonómii*, Opava: Silesian University, 2009. 242p. ISBN 978-80-7248-513-0.
6. MONTGOMMERY, D.C., JENNINGS, C.L., KULAHCI, M.: *Introduction to Time Series Analysis and Forecasting*, New Jersey: John Wiley & Sons, Inc, 2008. 445p. ISBN 978-0-471-65397-4.
7. ZHANG, G., PATUWO, B.E., HU, M.Y.: *Forecasting with Artificial Neural Networks*: *The State of Art*, International Journal of Forecasting 14, 1998. p35-p62.

Primary Paper Section: I

Secondary Paper Section: IN