Available online at www.sciencedirect.com

ScienceDirect

Procedia Economics and Finance 34 (2015) 194 – 201

Procedia Economics and Finance

www.elsevier.com/locate/procedia

Business Economics and Management 2015 Conference, BEM2015

Quantitative Modelling in Economics with Advanced Artificial Neural Networks

Lukas Falat[a]*, Lucia Pancikova[a]

[a]*Department of Macro and Microeconomics, Faculty of Management Science and Informatics, University of Žilina, Univerzitná 8215/1, 010 01, Žilina, Slovakia*

Abstract

In this paper, authors present a new approach in forecasting economic time series - application of artificial neural networks. Authors apply feed forward artificial neural network of the RBF type into the process of forecasting the financial data. Except for the standard RBF, authors also test their own new versions of this neural network combined with other techniques of the ML. These models represent new and more advanced version of the standard neural network. Authors add the evolutionary approach into the ANN and also combine the standard algorithm for adapting weights of the ANN with an unsupervised clustering algorithm called K-means. Finally, all of these methods are compared and contrasted with standard (statistical) approach on real economic data to show the potential of using artificial neural network in modelling economic variables.

Peer-review under responsibility of the Organizing Committee of BEM2015

Keywords: neural network, RBF, time series, economics, exchange rates

1. Introduction

Even though statistical economic time series forecasting started in the 1960s, the breakthrough came with publishing a study by Box and Jenkins (1976) where authors integrated all the knowledge about autoregressive and moving average models. From that time ARIMA models have been very popular in time series modelling for long time as O'Donovan (1983) showed that these models provided better results than other models used in that time. However, in 1982, Engle (1982) showed that using ARIMA models in financial modelling is not correct as these

* Lukas Falat. Tel.: +421 41 513 4425.
E-mail address: lukas.falat@fri.uniza.sk

Peer-review under responsibility of the Organizing Committee of BEM2015
doi:10.1016/S2212-5671(15)01619-6

series usually have conditional variance instead of constant. Therefore, he suggested ARCH (Autoregressive Conditional Variance) models for financial modelling.

However, more and more techniques of machine learning (ML) have started to be incorporated into the process of time series forecasting. One of reasons was the study of Bollershev (1986), where he proved the existence of nonlinearity in the financial data. One of the first ML techniques applied into time series forecasting were artificial neural networks (ANN). As ANN was a universal approximator, it was believed that these models could perform tasks like pattern recognition, classification or predictions (Anderson, 1988 and Hertz et al., 1991). Today, according to some studies such as Gooijer and Hyndman (2006), ANNs are the models having the biggest potential in predicting financial time series. The reason for attractiveness of ANNs for financial prediction can be found in works of [8], where authors showed that ANNs works best in connection with high-frequency financial data. While first applications f ANNs for financial forecasting, used the simplest feed forward ANN (perceptron) (White, 1988), nowadays it is mainly RBF (Orr, 1996) that is used in many research studies (Emam and Min, 2009; Qi, 2001; Yao and Tan, 2000) for this as it showed to be better approximator than the perceptron (Marček, 2004).

In this paper we will substitute the econometric model for economic predictions by the artificial neuron network prediction model. We decided to apply the (advanced) models of neural networks as multiple studies (Ntungo and Boyd, 1998; Boyaciouglu et al., 2009, Malliaris and Malliaris., 2009) showed that artificial neural network could perform better than standard statistical (econometrics) models.

In the first section of this paper, the machine learning methods are briefly described. In the section 3, the experiment (economic application) of suggested advanced neural networks is discussed. In section 4, the results are presented and discussed. Section 5 summarizes the paper.

2. Material and Methods

2.1. Artificial Neural Network

Mathematical model of the neuron was constructed on the base of functional neuron as a central element of human nervous system whose task is to transform information from one neuron to the others. The goal of mathematical neuron is a process identification. In other words, we try to find an input-output function so that the output would have desired parameters and the predicted error would be minimal. Let F: $x_t \in R^k \rightarrow y_t \in R^1$ be a projection assigning k-dimensional vector of inputs $x_t^T = (x_{1t}, x_{2t}, ..., x_{kt})$ one dimensional output y_t in specific time moment t. Let G: $G(x_t, w_t) : x_t \in R_{train}^k \rightarrow y_t \in R_{train}^1$ 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 known sample as it is possible. Let E(w) is function defined as

$$E(w_t) = \sum_{x_t, y_t \in R_{train}^k} (G(x_t, w_t) - y_t) \tag{1}$$

This function will be 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. Training or adaption is performed on training set. Validation set is used for validation of training network.

2.2. Radial Basis Neural Network

Radial Basis Function (RBF) neural network is an upgrade of multilayer perceptron network (MLP). The name comes from the name of its activation function. Generally, RBF is any real-valued function whose values depend only on the distance from the origin or from some other point c, called a center. Any function ϕ that satisfies this property is a radial function. The norm is usually Euclidean distance. Moreover, before providing predictions, the neural network of the RBF type must be adapted to approximate the data. Hence, the function defined in Eq. 1 must be minimal:

$$E(w_t) = \sum_{x_t, y_t \in R^k_{train}} (G(x_t, w_t, v_t) - y_t)^2 . \quad (2)$$

When E minimal, one can say the neural network represented by the function $G(x_t, w_t, v_t)$ is adapted to approximate the real function F. Implementation of $G(x_t, w_t, v_t)$ include two parts: counting of the neuron potential and activation of the neuron. Here lies the biggest difference between MLP and RBF –different functions for activating hidden neurons are used. RBF network uses radial basis function of Gaussian type instead of sigmoid function for activating neurons in hidden layer which is used at perceptron. The Gaussian function is defined for j^{th} hidden neuron (where σ_j^2 is the variance of j^{th} neuron and u is the potential of the neuron) as

$$\psi_1(u^j) = \exp(-u_j / 2\sigma_j^2) = \exp(- \| x - w_j \|^2 / 2\sigma_j^2) , j = 1, 2, ...s. \quad (3)$$

Finally, the network output for RBF neural network is counted as follows:

$$y = \psi_2(\sum_j v_j \psi_1(\| x - w \|)) = \sum_{j=1}^{s} v_j \exp(- \| x - w_j \|^2 / 2\sigma_j^2) \quad (4)$$

2.3. Genetic Algorithms

Genetic Algorithms (GA), which are machine learning algorithms for optimization, are stochastic search techniques that guide a population of solutions towards an optimum using the principles of evolution and natural genetics. Basic operators include reproduction, crossover and mutation.

Adopted from biological systems, genetic algorithms are based loosely on several features of biological evolution (Montana and Davis, 1989). They require five components (Davis, 1987):

1. A way of encoding solutions to the problem on chromosomes. In the original GA an individual chromosome is represented by a binary string. The bits of each string are called genes and their varying values alleles. A group of individual chromosomes are called a population.
2. An evaluation function which returns a rating for each chromosome given to it.
3. A way of initializing the population of chromosomes.
4. Operators that may be applied to parents when they reproduce to alter their genetic composition. Standard operators are mutation and crossover.
5. Parameter settings for the algorithm, the operators, and so forth.

Given these five components, a GA operates according to the following steps (Montana and Davis, 1989):

1. Initialize the population using the initialization procedure, and evaluate each member of the initial population.
2. Reproduce until a stopping criterion is met. Reproduction consists of iterations of the following steps:
   a) Choose one or more parents to reproduce. Selection is stochastic, but the individuals with the highest evaluations are usually favoured in the selection.
   b) Choose a genetic operator and apply it to the parents.
   c) Evaluate the children and accumulate them into a generation. After accumulating enough individuals, insert them into the population, replacing the worst current members of the population.

When components of the GA chosen appropriately, the reproduction process will generate better children from parents, converging finally on results close to a global optimum.

2.4. K-means clustering

K-means, which belongs to a group of unsupervised learning methods, is a non-hierarchical exclusive clustering method based on the relocation principle. It creates the optimum decomposition of objects into the previously

defined number of clusters. The method produces exactly k clusters so that the characteristic function used for clustering would be minimal. The most common type of characteristic function is location clustering. The characteristic function is then computed as the distance between the given object and a centroid (the centre of the cluster). The most common distance function is Euclidean. After decomposition of objects into clusters, new centroids are then counted. The process is repeated until the minimization function E defined in Eq. 5 converges

$$E = \sum_{j=1}^{k} \sum_{i=1}^{n} \| x_i^j - c_j \|^2 \tag{5}$$

where $\| x_i^j - c_j \|$ is the Euclidean distance between the input (x_i) and center of the nearest cluster c_j. In our experiments, we used the adaptive version of Kmeans [9] where coordinates of the centroids were adapted after every input vector according to the following formula

$$c_{j'} = c_{j'}^* + \eta(x_i - c_{j'}) \tag{6}$$

2.5. Application

Scientists incorporate other methods into RBF network in order to better its outputs. For example (Rivas, 2004) use GA for creating „Evolving" RBF – i.e. to automatically find the ideal number of hidden neurons. Kecman (2001) implements the soft and cloud concept into the RBF neural network. So the first hypothesis to be tested is that a combination of the standard RBF with an unsupervised learning method can be used to achieve better accuracy of the RBF neural network. Since (Kohonen, 1995) demonstrated that non-hierarchical clustering algorithms used with ANN can perform better results of the network, we will incorporate K-means into the RBF so as to find out whether this combination can produce the effective improvement of the standard RBF in the domain of financial time series.

Moreover, in recent years GA have become a popular optimization tool. Therefore, the standard backpropagation (BP) (which is very often considered a weakness of RBF) will be substituted by the GA as an alternative learning technique in the process of weights adaptation.

2.6. Data and Model Validation

To implement artificial neural network into the prediction process of economic variables as well as to test our hypothesis we used high-frequency time series data - daily close prices of the USD/CAD currency pair. The interval was from 10/31/2008 to 10/31/2012, i.e. 1044 daily observations. The data was downloaded from a website http://www.global-view.com/forex-trading-tools/forex-history. Due to validation of a model, data were divided into two parts. The first part included 912 observations (from 10/31/2008 to 4/30/2012) and was used for training of the model. The second part of data (5/1/2012 to 10/31/2012) counting 132 observations, was used for model validation by making static one-day-ahead ex-post forecast. These observations were not incorporated into model training, so parameters of a model were not changing anymore. This was done in order to find out the prediction power of model as there is an assumption that if the model can handle to predict ex-post data, it will also be able to perform real predictions.

2.7. Box-Jenkins Analysis

Box-Jenkins analysis was performed to make a comparison between standard statistical models and our neural network models. For statistical modelling Eviews software was used. By analysing autocorrelation and partial autocorrelation function of the first differences we found out that there was no strong dependency between random parts of the model. Therefore, the model was identified as AR(0). By analysing residuals and squared residuals using

Ljung-Box test and ARCH test (Engle, 1982) we found out that ARCH effect was present in residuals. Due to that, the residuals were modelled by various ARCH and GARCH models as well as GARCH extensions (PGARCH, EGARCH, TGARCH). We used Berndt-Hall-Hall-Hausman optimization for finding the optimal values of GARCH parameters; initial values of parameters were counted using Ordinary Least Squares and these values were then by iterative process consisted of 500 iterations. Convergence rate was set to 0.0001. Finally, we tested standardized residuals with Ljung-Box Q test to confirm there were no significant coefficients in residuals of the model. The assumption was confirmed, so according to statistical tests the model was correct. Finally, AR(0) + EGARCH(1,1,1) with Gaussian error distribution was chosen as the model with best evaluation characteristics (MSE). The model is defined as follows

$$y_t = e_t \sqrt{h_t} \tag{7}$$

$$\log(h_t) = -0{,}172109 + 0{,}117148 \frac{|\varepsilon_{t-1}|}{\sqrt{h_t}} + 0.037398 \frac{|\varepsilon_{t-1}|}{\sqrt{h_t}} + 0{,}992135 \log(h_{t-1}) \tag{8}$$

2.8. Neural Network Settings

We used own application of RBF network implemented in JAVA with one hidden layer where we tested from three to ten processing neurons to achieve best results of network. For every model, only the result with the best configuration is stated. We used the identity function as an activation function for the output layer too. The weights of network were initiated randomly – generated from the uniform distribution <0,1). As for the back-propagation, the learning rate was set to 0.001 to avoid the easy imprisonment in local minimum. The number of epochs for each experiment with BP was set to 5000 as this showed to be a good number for backpropagation convergence. The final results were taken from the best of 5000 epochs and not from the last epoch in order to avoid overfitting of the neural network. As we used non-standardized data, we analysed original series for autocorrelation. As there was a strong dependence on the previous day (0,996) we used just one network input - the previous observation.

2.9. K-means clustering

We used K-means in the phase of non-random initialization of weight vector *w* performed before the phase of network learning performed by backpropagation. Cluster ordinances were initiated as ordinances of randomly chosen input vector. After, every input vector was assigned the nearest cluster. When done, the coordinates of clusters were recounted. This cycle was repeated 5000 times and the learning rate for cluster adaptation was set to 0.001. The number of clusters was set to the number of hidden neurons.

2.10. Genetic Algorithm

Our own implementation of the genetic algorithm we used for weight adaptation. The chromosome length was set according to the formula:

$$CL = D * s + s, \tag{9}$$

where s is the number of hidden neurons and D is the dimension of the input vector. A specific gene of a chromosome was a float value and represented a specific weight in the ANN. The whole chromosome represented weights of the whole network. The fitting function for evaluating the chromosomes was the mean square error function. The chromosome with the best MSE was automatically transferred into the next generation. The other individuals of the next generation were chosen as follows: by tournament selection (size of the tournament equalled to 100) 100 individuals were randomly chosen from the population. The fittest of them was then chosen as a parent. The second parent was chosen in the same way. New individual was then created by the crossover operation. If the generated value from <0,1) was lower than 0.5, the weight of the first parent at the specific position was assigned to the new individual. Otherwise, the new individual received the weight of the second parent.

The mutation rate was set to 0.01. If performed, the specific gene of a chromosome was changed to a random value. The size of the population and the number of generations was set accordingly to the settings of BP. In BP,

there were 5000 cycles of the forward signal propagation plus 5000 cycles of backward error propagation. In GA we used the size of the population equalled to 1000 and 10 as the number of generations.

3. Results and Discussion

We used MSE (Mean Squared Error) numerical characteristic for assessing models. The results in Table 1 of a given model is from the best neuron configuration (in every model we tested number of hidden neuron from 3 to 10 to find the best output results of the network). Experiment for every model configuration was performed 12 times; the best and worst results were eliminated and from the rest the mean and standard deviation were counted.

Table 1. Prediction accuracy of created artificial neural networks (measured by MSE on ex-post set)

Regressor	NNHL	Neural Network	MSE	Standard Deviation
AR(1)		Standard	0.0000282628	0,0000129939
	3	ANN with K-means	0.0000175381	0.0000006224
		ANN with GA	0.0000180929	0.0000016469
		Standard	0.0000183763	0.0000028765
	4	ANN with K-means	0.0000173006	0.0000004025
		ANN with GA	0.0000176860	0.0000006219
		Standard	0.0000299369	0.0000812952
	5	ANN with K-means	0.0000174326	0.0000007575
		ANN with GA	0.0000176925	0.0000016246
		Standard	0.0000248756	0.0000105719
	6	ANN with K-means	0.0000187115	0.0000024836
		ANN with GA	0.0000205995	0.0000073265
		Standard	0.000029955	0.0000381995
	7	ANN with K-means	0.0000170959	0.0000002617
		ANN with GA	0.0000265817	0.0000100553
		Standard	0.0000530843	0.0000462909
	8	ANN with K-means	0.0000169521	0.0000003200
		ANN with GA	0.0000181709	0.0000016133
		Standard	0.0000594814	0.0000611668
	9	ANN with K-means	0.0000168649	0.0000002319
		ANN with GA	0.0000290958	0.0000136948
		Standard	0.0000842809	0.0000580551
	10	ANN with K-means	0.0000179805	0.0000029834
		ANN with GA	0.0000236821	0.0000093964

AR(1) autoregressive process of the order one,
NNHL number of neurons in the hidden layer
BP back-propagation, *KM* K-means clustering, *GA* genetic algorithm

Table 2 Final comparison of predictive qualities – best configurations (out-of-sample predictions measured by MSE on ex-post)

Model	Model Optimization	mean MSE	Standard deviation
Neural Network (Standard)	Back-propagation	0,0000183763	0,0000028765
Neural Network (with K-means)	*K*-means + Back-propagation	0,0000168649	0,0000002319
Neural Network (with GA)	Genetic algorithm	0,0000176860	0,0000006219
AR(0)- EGARCH(1,1,1)	Berndt-Hall-Hall-Hausman	0.0000170651	-

Looking in the table 1, our hypothesis were confirmed. The RBF network combined with K-means or GA in the optimization process provided significantly better results than the original RBF. ANN with K-means bettered the accuracy of prediction by 57.38 per cent while the ANN with GA better the accuracy of the predictions by 47.72 per cent. Moreover, besides lower MSE error, another advantage of using GA or K-means upgrade in economic prediction of the neural network is the consistency of predictions that could be seen in the results of the standard deviation. The standard deviation of the BP+K-means combination was (compared to the standard ANN) lowered by than 97.41 per cent. The standard deviation of genetic algorithm compared to the standard ANN was lowered by 85.23 per cent.

When comparing weights adaptation via GA and K-means plus BP, the results are almost the same. The prediction accuracy was the lowest at K-means + BP (18.48 per cent lower compared to genetic algorithm), whereas the standard deviation (consistency of predictions) was lower when using genetic algorithm. However, on base of our experiments we are of the opinion that GA has a bigger potential to perform even better forecasts as there are more parameters had to optimized. Additionally, backpropagation, even though used with K-means, seemed to reach its global minimum as even with the higher number of epochs (we tested 10000) the results were almost the same. The strength of K-means is the speed of convergence of the network. Without K-means, it took considerably longer time to achieve the minimum. If the K-means was used for setting the weights, the time for reaching the minimum was much shorter. So following from that, in many cases it is not necessary to interpolate the output value by radial functions, it is quite sufficient to use one function for a set of data (cluster), whose centre is considered to be a centre of activation function of a neuron and the values of centroids can be used as an initialization values of weight vector w. The advantage is that lower number of epochs is supposed be used for network training. Moreover, K-means is also simple to implement. However, one must bear in mind that it is efficient only in the domain of non-extreme values. Otherwise, other non-hierarchical clustering algorithms must be used.

As for GA, the convergence is also considerably faster than at BP. As for GA, the convergence is also considerably faster than at backpropagation. In most cases the GA algorithm converged in 5 generation what is 50 per cent faster than the standard back-propagation. In addition, GA does not have the same problem with scaling as BP as it generally improves the current best candidate monotonically. It does this by keeping the current best individual as part of their population while they search for better candidates. Moreover, supervised learning algorithms suffer from the possibility of getting trapped on suboptimal solutions. GA are generally not bothered by local minima. The mutation and crossover operators can step from a valley across a hill to an even lower valley with no more difficulty than descending directly into a valley. The disadvantage of using GA in the ANN is that it demands to set up a lot of parameters.

Finally, when comparing the neural network models results with standard statistical Box-Jenkins models which are mainly used for economic predictions, one can state that the RBF+K-means neural network model performed better predictions that the standard AR(0)+EGARCH(1,1,1) model. However, it is important to note that this paper did not focus primary on comparing the neural network models with statistical models. It rather focused on comparing the neural network models between themselves – to test the implemented advanced neural network with the standard RBF. The parameters of these algorithms were set to the comparable scale and the results of these algorithms are not the optimal values as of statistical models. We believe (and there is a reasonable assumption) that if we used more replication in the optimization process of the neural network models (f. ex. more than ten generations in the process of forming weights of ANN via GA), predictions would be more accurate. However, in that case the comparable part between GA and BP would be lost.

4. Conclusion

In this paper we performed economic time series predictions with various models of neural networks. We used USD/CAD data which was later divided into training set and validation due to model checking. Except for a standard ANN of the RBF type, we also combined an unsupervised learning method called K-means and GA into the RBF in order to achieve better accuracy of the network. Both of the algorithms were used in the process of adapting weights (optimization process) of the network. The reason for incorporating other algorithms into the network was that the BP is considered a weakness of the RBF. Some of drawbacks of BP include scaling problem, complexity problem, slow convergence, convergence to a local minimum etc. Therefore K-means algorithm was used in the phase of non-random initialization of weight vector w before the phase of network learning. Moreover, we also eliminated the BP in the second part of our experiment and used GA instead. The final comparison of the

selected methods for weight adaptation was performed and both of these upgrades showed to be helpful in the process of creating better forecasts. The main advantage of these advanced methods was the accuracy of the predictions.

Finally, to sum up, we performed and implemented various models of advanced artificial neural networks and showed an alternative way in making accurate economic predictions of various economic variables. The accuracy of this approach showed to be on the comparative scale with standard models. However, except for this, this approach has many other advantages such as more flexibility, automatization, black-box approach etc. We therefore believe that this approach has definitely a big potential as it represents the modern tool of using ICT in business with many advantages.

Acknowledgements

This paper was supported by VEGA grant project 1/0942/14: Dynamic modelling and soft techniques in prediction of economic variables.

References

Anderson, J.A., Rosenfeld, E., 1988. A collection of articles summarizing the state-of-the-art as of 1988. Neurocomputing: Foundations of Research, MIT Press.

Bollershev, T., 1986. *Generalized Autoregressive Conditional Heteroskedasticity.* Journal of Econometrics 31, 307–327.

Box, G. E. P., Jenkins, G.M., 1976. Time series analysis: forecasting and control. Holden-Day, San Francisco, CA.

Boyacioglu, M.A, Yakup, K., Omer, K. B., 2009. Predicting Bank Financial Failures Using Neural Networks, Support Vector Machines and Multivariate Statistical Methods: A Comparative Analysis in the Sample of Savings Deposit Insurance Fund (SDIF) Transferred Banks in Turkey, Expert Systems with Applications 36, 3355 – 3366.

Davis, L., 1987. Genetic Algorithms and Simulated Annealing. Pitman, London.

Emam, A., Min, H., 2009. The artificial neural network for forecasting foreign exchange rates, International Journal of Services and Operations Management 5(6), 740 – 757.

Engle, R. F. 1982. Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of United Kingdom Inflation. Econometrica 50, 987-1008.

Gooijer, J. G., Hyndman, R. J., 2006. 25 years of time series forecasting. International Journal of Forecasting 22, 443-473.

Hertz, J., Krogh, A., Palmer, R. G., 1991. Introduction to the Theory of Neural Computation. Westview Press.

Hill, T. L., Marquez, M., O'Connor, Remus, W., 1994. Neural networks models for forecasting and decision making. International Journal of Forecasting 10.

Kecman, V., 2001. Learning and Soft Computing – Support Vector Machines, Neural Networks and Fuzzy Logic Models, Massachusetts Institute of Technology.

Kohonen, T., 1995. Self-organising maps. Springer, New York.

Malliaris, M., Malliaris, A. G., 2009. Modeling Federal Funds Rates: A Comparison of Four Methodologies, Neural Computing and Applications 18, 37 – 44.

Marček, D., 2004. Some Intelligent Approaches to Stock Price Modelling and Forecasting. Journal of Information, Control and Management Systems 2.

Montana, D. J., Davis, L., 1989. Training Feedforward Neural Networks Using Genetic Algorithms.

Ntungo, C., Boyd, M., 1998. Commodity Futures Trading Performance Using Neural Network Models versus ARIMA Models, Journal of Futures Markets 18, 965 – 983.

O'Donovan, T. M., 1983. Short Term Forecasting: An Introduction to the Box-Jenkins Approach. NY: New York, Wiley.

Orr, M. J. L., 1996. Introduction to Radial Basis Function Networks.

Qi, M., 2001. Predicting US recessions with leading indicators via neural network models. International Journal of Forecasting 17, 383– 401.

Rivas, V. M., Merelo, J. J., Castillo, P. A., Arenas, M. G., Castellano, J. G., 2004. *Evolving RBF neural networks for time-series forecasting with EvRBF.* Information Sciences 165, 207-220.

White, H., 1988. Economic prediction using neural networks: the case of IBM daily stock returns. Proceedings of the 2nd IEEE conference on neural networks, Vol. 2, pp. 451–458.

Yao, J., Tan, C. L., 2000. A Case Study On Using Neural Networks to Perform Technical Forecasting on Forex, Neurocomputing 34, 79 – 98.