

Genetic Algorithm Neural Network model vs Backpropagation Neural Network model for GDP Forecasting

Dezdemonja Gjylapi
Computer Science Dept.
University "Ismail Qemali",
Vlorë
dezdemonja.gjylapi@univlora.edu.al

Eljona Proko
Computer Science Dept.
University "Ismail Qemali",
Vlorë
eljona.proko@univlora.edu.al

Alketa Hyso
Computer Science Dept.
University "Ismail Qemali",
Vlorë
alketa.hyso@univlora.edu.al

Abstract

This paper evaluates the usefulness of neural networks in GDP forecasting. It is focused on comparing a neural network model trained with genetic algorithm (GANN) to a backpropagation neural network model, both used to forecast the GDP of Albania. Its forecasting is of particular importance in decision-making issues in the field of economy. The conclusion is that the GANN model achieves higher accuracy on GDP forecasting. AG improves ANN model performance compared with standard backpropagation ANN model.

1. Introduction

The quarterly GDP data are important for economic analysis, because it gives insight on the general economic activity, on the fluctuations of business cycle and on the economic turning points. Forecast of macroeconomic variables such as GDP play an important role in monetary policy decisions and in assessing the future economic situation. Economic policymakers and analysts can adapt their theoretical analysis of economic conditions according to the forecasts of macroeconomic variables, or even perhaps use them as a support and an explanation of their theoretical analysis. Forecasts with better performance on macroeconomic variables will lead to better decisions.

To have an assessment of the economic situation of the country, policy institutions need to have information on the indicator of GDP, in order to analyze the macroeconomic policies implemented in the past and to take political decisions about the future.

Tkacz and Hu (1999) [TKA99] studied forecasting GDP of Canada. The result of this study is that the best

neural network models outperform the best linear models by between 15 and 19 per cent at this horizon, implying that neural network models can be exploited for noticeable gains in forecast accuracy.

Giovanis (2009) [GIO09] in his paper is using the ARIMA and ANN to predict the rate of economic growth in the USA. This study examines the estimation and forecasting performance of ARIMA models in comparison with some of the most popular and common models of neural networks. The results of this study indicate that neural networks models outperform the ARIMA forecasting.

(Çeliku, Kristo and Boka) (2009) in the discussion paper treats several models to forecast quarterly GDP in Albania. They consist on ARIMA models with seasonal components and indicator models, similar to bridge models. This paper presents a first attempt to model the GDP using a multi equations system which accounts for the sectional interactions [ÇEL09].

Models that predict GDP in the short term use surveys' indicators because these indicators have several advantages such as:

1. Provide preliminary signals about short-term developments of the activities of economic agents.
2. These indicators are published formerly than the main macroeconomic aggregates
3. These indicators are rarely subject to adjustments.

Due to the difficulties encountered in modeling the quarterly GDP and the fact that ANN has proven to be an efficient tool for non-parametric model data in the form of non-linear function; we have developed a model for the Albania's Gross Domestic Product forecasting with artificial neural network approach. In this paper, we forecast the GDP using a neural network model trained by genetic algorithm (GANN) and compare it with a

backpropagation neural network model, also used to predict the Albania's GDP.

The rest of the paper is organized as follows. Section 2 presents a brief description of the ANN for GDP forecasting, Section 3 explains the details of the GANN model, Section 4 explains the results of the GANN and BPNN models, and finally, Section 5 presents our conclusions.

2. Artificial Neural Networks for GDP Forecasting

The advantage of artificial neural network approach is that this model can capture the relationship of nonlinear data, especially if the economy is very volatile, and it is superior when forecasting chaotic data [IMA11].

An ANN consists of a number of simple and interconnected processors, also called neurons, analogous to biological neurons in the brain. Each neuron receives a number of input signals through its connections, however, it gives no more than one output signal. The output signal is transmitted through the neuron's exit line [NEG05].

Developing an ANN comprises the definition of:

- 1 - the network architecture, which is defined by the basic processing elements (i.e. neurons) and by the way in which they are interconnected (i.e. layers);
- 2 - the NN Learning, which implies that a processing unit is capable of changing its input or output behavior as a result of changes in the environment, i.e. to adjust the weights based on input vector values;
- 3 - the data used for training, testing and validating the neural network.

In this paper two neural networks have been implemented: the multilayer perceptron (MLP, Figure 1) neural networks trained by back-propagation (BP), and the MLP neural network trained by a genetic algorithm (GANN), both using bipolar sigmoid activation function, which form is as follows:

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

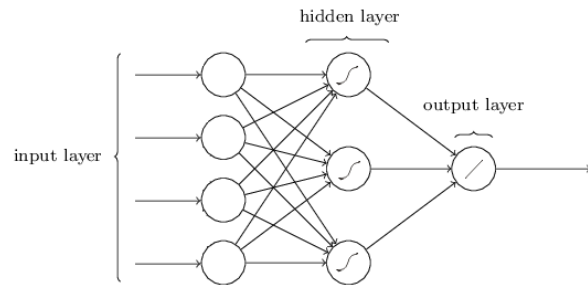


Figure 1. The structure of a MLP network

2.1 The data used for the neural network

The accuracy of GDP forecasting with ANN depends on the selection of the variables to include as input to the network.

In this selection process we were based on the paper of Celiku et al.[ÇEL09] to use the variables judging their potential economic correlations with the quarterly GDP.

In this paper we use all these economic and financial variables combined with surveys' variables.

a. Economic variables

- Government expenditures, data provided by the Ministry of Finance.
- Construction permissions for residential and business purposes, in value data provided by INSTAT.
- Total imports, data provided by the INSTAT.

b. Financial variables

- Interest rate on loans denominated in EURO, data provided by the Bank of Albania;

c. Variables from surveys

Variables from surveys are based on indicators constructed from qualitative surveys developed by the Bank of Albania with businesses and consumers.

- ESI (Alb. TNE), the Economic Sentiment Indicator aggregates in a single indicator the opinions of the main market agents. They are collected from the confidence surveys for the industry, construction and services sector and for the consumers.
- EI (Alb. TE) the survey indicator for the economy
- II (Alb. TI) the survey indicator for industry sector
- CI (Alb. TN) the survey indicator for construction sector

- SI (Alb.TSH) the survey indicator for services sector
- MPI (Alb. TBM) the survey indicator for major purchases

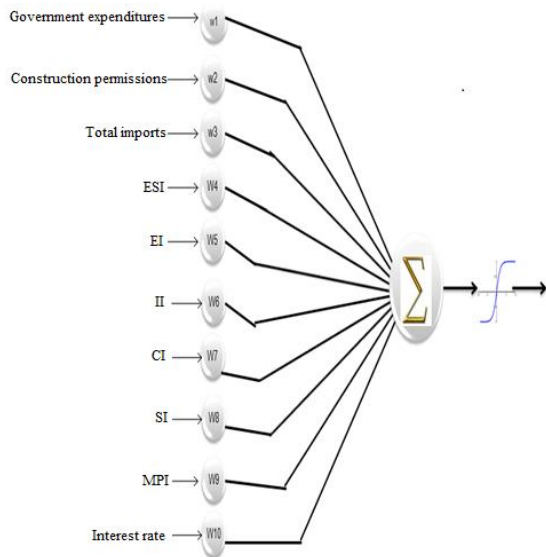


Figure 2. The model of a neuron in the hidden layer

The Figure 2 presents the sample of a neuron of the hidden layer for the GDP forecasting, used in both models, GANN and BPNN.

The data used to forecast GDP are quarterly data from 2002Q2 till 2016Q1. These data was stored in a .csv file and served as input to the neural network models. In general, ANN models require large amounts of data. For the case of Albania’s GDP forecasting, the biggest inhibitor is the lack of sufficient data. This encompasses availability of data, its consistency and the data time span.

2.2 BPNN for Albania’s GDP Forecasting

In a GDP forecast by using the neural network trained with the Backpropagation method, it was used a three-layer architecture, i.e. the input, output, and the hidden layer. The input layer, similarly as in the “neuro-genetic” model, it has 10 neurons, the hidden layer 20,

and the output layer has only one. The main parameters of the BP used in BPNN model were set as:

Momentum = 0.5
Learning rate = 0.3

The results are presented in Figure 3.

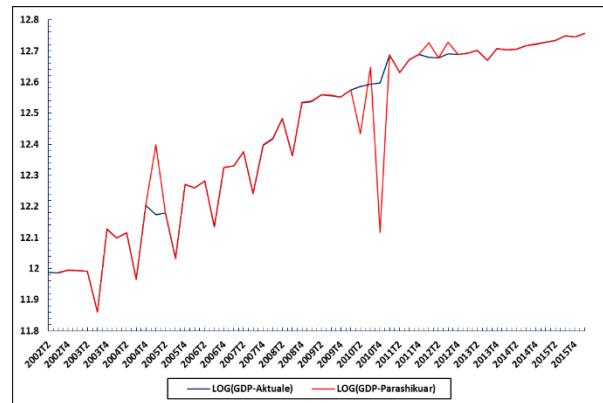


Figure 3. LOG(GDP) vs LOG(GDP forecasted) using BPNN

In Table 1 are presented the results of an evaluation performed for calculating the accuracy of the forecast.

Table 1: Indicators of the accuracy of the GDP forecast with Backpropagation NN.

Indicator	Value
MFE ¹	0.0047
MAD ²	0.0184
MSE ³	0.0056
TS ⁴	0.0052

In Table 1 are presented the results of an evaluation performed for calculating the accuracy of the forecast.

¹ MFE-Mean Forecasted Error

² MAD-Mean Absolute Deviation

³ MSE - Mean Squares Error

⁴ TS - Tracing Signal

Table 1 shows that this model works, as $TS = 0.0052$, and since the $MFE > 0$ the model tends to under-forecast, with a mean square error of 0.0056 units.

3. GANN Model for Albania's GDP Forecasting

3.1 Network architecture

In this model is used the artificial neural network Feedforward with multiple layers. Several tests are conducted with networks with 3, 4, and 5 layers. In the three cases the first layer, which is the input layer, it has 10 neurons, the same number of the factors which are determined to be influencers in the forecast of the GDP. It is the same with the last layer, the output one, which consists of only one neuron.

The hidden layers have a different number of neurons: the 3-layer model has a hidden layer consisting of 20 neurons (two times the number of neurons of input layer); the 4-layer model has two hidden layers with respectively 20 and 10 neurons; while the 5-layer model has three hidden layers with respectively 20, 20, and 10 neurons.

3.2 The learning Algorithm of the GANN model

There are two types of learning algorithms: the gradient descent and the global search method. The methods such as Backpropagation, Newton, Quasi-Newton, and Levenberg - Marquand can be classified as gradient descent methods. While the genetic algorithm is a global search method.

ANN model is very useful, especially in the cases where the process of data generation is nonlinear and complex, or when the functional form is not clear. However, the learning process or the adjustment of the parameters in an ANN model can take time and can fall also in a trap, such as the local minima, especially when there are too many parameters. To avoid such potential problems, we can use the GA in the learning process.

The advantage of the ANN model with genetic algorithms is that this model can capture the relationship of the nonlinear data, especially if the economy is very unstable; and it is a superior model when it comes to predict chaotic data.

There are four main differences to be distinguished in what respects genetic algorithms (GAs) differ from

traditional search and optimization procedures that make them such a robust method [GOL92]:

- 1) GAs use an encoding of the parameters, not the parameters themselves;
- 2) GAs search from a population of search points, not a single point;
- 3) GAs only use the objective function to judge solution quality, not derivatives or other auxiliary knowledge;
- 4) GAs use probabilistic transition rules, not deterministic rules.

The basic idea in GA is finding the suitable individual in the current population. In this context, the GA searches for the vector of coefficients which is the global optimal solution. The basic process of the GA is represented as follows. It starts with an initial population selected in a random manner. If the population converges, the process stops and the solution is presented. Otherwise, new individuals are created from the old ones and after some operations the new generation is created. This process is repeated until the objective is reached, explained in more details as follows:

The objective of GA is to minimize the sum of squares error (SSE), subject of the parameters generated by the algorithm, i.e.

$$\text{Min} \sum (y_i - \bar{y}_i)^2 \quad \text{ku} \quad \bar{y} = f(x | \theta)$$

Where \bar{y} is the output generated from the model, and θ are the coefficients parameters, which minimize the error function. GA has 8 main steps, which are described as follows:

- 1) The creation of an initial population of vectors with coefficients $[\theta_1, \theta_2, \dots, \theta_p]$, where p is an even number, and θ_i is a vector of $K \times 1$ elements. The initial population can be created randomly from the normal standard distribution, or using constrains on the sign or interval of the parameters' values.
- 2) Two couples are selected randomly from the initial population. The fitness of these four vectors is estimated in regards to the objective function and two vectors with the best fitness (smallest SSE) are selected as winners. These two vectors are also referred to as parents.

- 3) Through the application of the crossover operator from the parent vectors are created two new vectors, called *children*. The simplest form of the crossover operator is the one-point crossover, through which the two parents intersect at point I and the parts on the right of point I switch the positions. I is selected randomly from the set $(1, K-1)$. For $K=6$ and $I=3$, the operator would turn out as below:

$$\begin{bmatrix} \theta_{11} \\ \theta_{12} \\ \theta_{13} \\ \text{---} \\ \theta_{14} \\ \theta_{15} \\ \theta_{16} \end{bmatrix}_{P1} \quad \begin{bmatrix} \theta_{21} \\ \theta_{22} \\ \theta_{23} \\ \text{---} \\ \theta_{24} \\ \theta_{25} \\ \theta_{26} \end{bmatrix}_{P2} \Rightarrow \begin{bmatrix} \theta_{11} \\ \theta_{12} \\ \theta_{13} \\ \text{---} \\ \theta_{24} \\ \theta_{25} \\ \theta_{26} \end{bmatrix}_{F1} \quad \begin{bmatrix} \theta_{21} \\ \theta_{22} \\ \theta_{23} \\ \text{---} \\ \theta_{14} \\ \theta_{15} \\ \theta_{16} \end{bmatrix}_{F2}$$

- 4) The mutation operator is applied to each element of the children vectors. Under the influence of this operator, each of these elements is subject of a hit with a low probability, $\mu > 0$. The probability is typically given by the formula: $\mu = 0.15 + 0.33 / G$, where G is the size of the generation, while in our application it is defined by the user. The mutation operator looks like the following [MIC96]:

$$\theta = \begin{cases} \theta + s[1 - r_2^{(1-t/T)^b}] & \text{if } r_1 > 0.5 \\ \theta - s[1 - r_2^{(1-t/T)^b}] & \text{if } r_1 < 0.5 \end{cases}$$

While in our application we use the following mutation operator:

$$\theta = \begin{cases} \theta + r_2 & \text{if } r_1 \geq 0.5 \\ \theta * r_2 & \text{if } r_1 < 0.5 \end{cases}$$

Where r_1, r_2 are two real numbers from the range $[0, 1]$, selected randomly and s is a random number from a normal standard distribution, t – the number of the next generation, T – the maximal number of the generations, and b is a parameter which defines the degree in which the mutation operator is non-uniform.

- 5) A “fight” runs between the four individuals, the parents and the children (P1, P2, F1 and F2). The two best vectors, the ones with the lowest sum of squares error will survive, and they will move to the next generation, whereas the two others will be eliminated.
- 6) The process repeats, returning the parents again in the pool of the population, so that they have the possibility of re-selection, until the next generation is populated with P vectors of coefficients.
- 7) The members of the current generation are evaluated together with the ones of the new generation, in regards to the fitness criteria, and the best one for the next generation is selected. Hence, the concept of “Elitism” is applied.
- 8) Create new generations of populations with P individuals and assess the convergence through the behavior of the best member in each generation, based on the fitness criteria. If the change in the assessment of fitness of the best member in each generation, which passes through 50 generations, is small, it can be claimed that genetic research has converged on an optimum.

The GANN learning flowchart is presented in Figure 4.

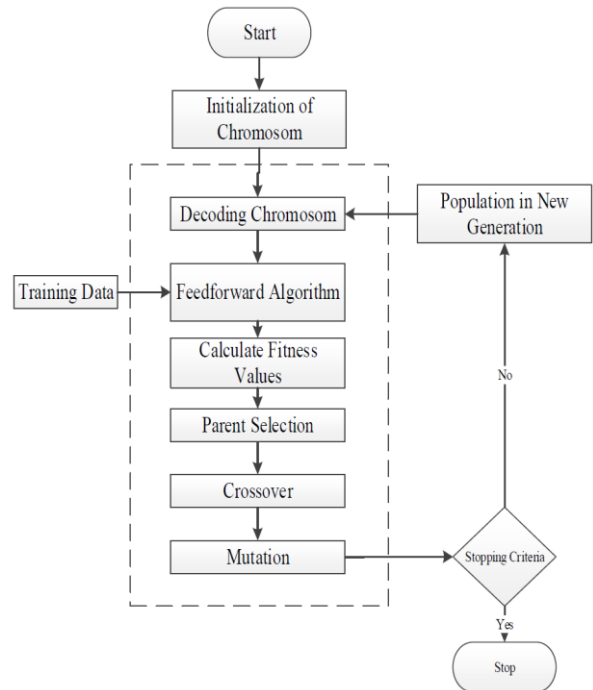


Figure 4. NN learning using GA (GANN) [NUR14]

3.3 GANN model results

The main parameters of the GA used in GANN model were set as:

- ✓ Genetic population Size = 100
- ✓ Crossover probability in genetic population = 0.6
- ✓ Mutation probability in genetic population = 0.4
- ✓ Probability to add newly generated chromosome to population = 0.25
- ✓ The bipolar sigmoid coefficient $\alpha = 1.5$

We tested three types of architectures. The results are presented in Table 2

Table 2: Indicators of the accuracy of the GDP forecast with different GANN architectures.

Indicator	3-layers	4-layers	5-layers
MFE	0.0006	0.0005	0.0005
MAD	0.0087	0.0081	0.0082
MSE	0.0004	0.0004	0.0005
TS	3.9555	3.6100	3.4870

As shown in Table 2, it results that the model with 4-layers architecture is the best model. The results of this model are presented in Figure 5.

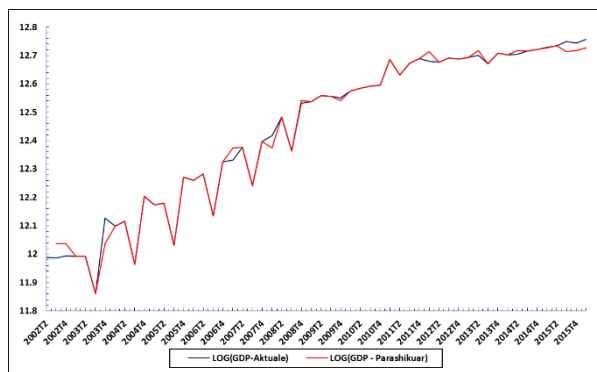


Figure 5. 4-layers GANN model forecasting

Table 3: Indicators of the accuracy of the GDP forecast with GANN.

Indicator	Value
-----------	-------

MFE	0.0005
MAD	0.0081
MSE	0.0004
TS	3.6100

Table 3 shows that this model works, as $TS = 3.61$, and since the $MFE > 0$ the model tends to under-forecast, with a mean square error of 0.0004 units.

4. GANN model vs BPNN Model

We compare BPNN and the GANN model using the indicators of accuracy. Details about the comparison are shown in Table 4.

Table 4: Indicators of the accuracy of the GDP forecast with BPNN and GANN.

Indicator	BP	GANN
MFE	0.0047	0.0005
MAD	0.0184	0.0081
MSE	0.0056	0.0004
TS	0.0052	3.6100

5. Conclusions

In the model developed in this paper it is treated exactly the evolution of neural network weights through genetic algorithm.

Due to the simplicity and generalization of evolution and the fact that training algorithms based on gradient often need to be executed several times in order to avoid being trapped in a local minima, the evolution technique is highly competitive.

The results show that the model that uses neural networks to forecast GDP, regardless of the method used for weight training, it works and has a very satisfactory performance.

As long as the tracking signal (TS) is between -4 and 4 , (in our model TS is equal to $GANN=3.61$ and $BPNN=0.0052$) we can say that the model is working correctly.

The GANN forecasted GDP in this study resulted with a MSE equal to 0.0004 while the BPNN forecasted GDP resulted with a MSE equal to 0.0056.

The GANN model tends to slightly under-forecast, with an average absolute error of 0.0081 units, while the BPNN model with an average absolute error of 0.0184 units.

GANN outperforms better than BPNN in Albania's GDP forecasting.

References

- [ÇEL09]Çeliku, Evelina, Ermelinda Kristo, and Merita Boka. 2009. "Modelimi i PBB-së tremujore. Roli i treguesve ekonomikë dhe atyre të vrojtimeve." Tirane: Banka e Shqipërisë.
- [GIO09] Giovanis, Eleftherios. 2009. "ARIMA and Neural Networks. An application to the real GNP growth rate and the unemployment rate of U.S.A." *SSRN Electronic Journal*. doi:10.2139.
- [GOL92]Goldberg, David E. 1992. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing.
- [IMA11]Imansyah, Muhammad Handry, Suryani, Nurhidayat, and Muzdalifah. 2011. "GDP Estimation and Slow Down Signal Model for Indonesia: An Artificial Neural Network Approach." *Finance and Banking Journal* 13 (1): 77-94.
- [MIC96]Michalewicz, Zbigniew. 1996. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag Berlin Heidelberg.
- [NEG05]Negnevitsky, Michael. 2005. *Artificial Intelligence: A Guide to Intelligent Systems*. Pearson Education.
- [NUR14]Nurcahyo, Septian, Fhira Nhita, and Adiwijaya. 2014. "Rainfall Prediction in Kemayoran Jakarta Using Hybrid Genetic Algorithm (GA) and Partially Connected Feedforward Neural Network (PCFNN)." *2nd International Conference on Information and Communication Technology (ICoICT)*.
- [TKA99]Tkacz, G. and S. Hu. 1999. "Forecasting GDP growth using artificial neural networks." Bank of Canada WP, No 99.