GROWTH RATE PREDICTION USING ANN IN PRODUCTION MANAGEMENT

Yılmaz GÖBENEZ*

ÖZET

Büyüme oranının tahmini ekonomi yönetimi, üretim yönetimi ve bütçe planlama için çok önemlidir. Bu amaçla hem zaman serileri hem de Yapay Zeka Teknikleri kullanılabilir. Bu teknikleri her ikisini de kullanarak böyle bir çalışma bu makale de yapılmıştır. Zaman serisi verisi ile yıllık büyüme oranı öğrenilebilir ve böylece değişkenlerin geçmiş değerleri gelecek değerleri tahmin etmekte kullanılabilir.

Bu makalede, yıllık büyüme oranı, bir otoregresiv model (AR) ve bir Yapay Sinir Ağları (ANN) modeli yardımıyla ANN simülatörü kullanılarak gerçekleştirilmiştir.

Gelecek yıllarda büyüme oranları , ANN simülatörü ve zaman serisi analizi yardımıyla yukarıda bahsedilen parametrelere bağlı olarak tahmin edilmiştir. Sonra sonuçlar karşılaştırılmıştır. Sonuçlar göstermiştir ki ANN model yardımıyla elde edilen tahmin hatası AR modeli ile elde edilenden daha küçüktür.

Anahtar Kelimler: Büyüme Oranı, Üretim Yönetimi, Yapay Sinir Ağları , Tahminleme, Zaman Serisi

ABSTRACT

The prediction of growth rate is a vital importance for economy management, production planning and budget planning. Both the time series techniques and artificial intelligent techniques can be used for this aim. Such a prediction study using both of these techniques is made in this study. With time series data, the annual growth rate can be learned and therefore past values of variables can be used to predict the future values.

In this paper, annual growth rate has been predicted by an autoregressive (AR) model and an artificial neural network (ANN) model implemented by ANN simulator developed by the researcher.

Future growth rates have been forecasted depending on the above mentioned parameters via time series analysis and ANN simulator then the results were compared. The results showed that the prediction error obtained by ANN model is smaller than the error obtained by AR model.

Keywords: Growth rate, Production Management, Artificial Neural Network, Forecasting, Time Series.

1. INTRODUCTION

Turkey had been out of international competition and governed by closed economy rules until January 24th 1980. The country adapted the liberal economy principles through what is known as "January 24th Decisions". The main purpose of the so-called January 24th Decisions was to run the domestic economy based on free market economic principles and to realize integration with the world economy. Businesses producing in a free market economy experience growth difficulty for various reasons. Growth is very important for a business. Strong management can not be possible in a non-growing business.

* Doç. Dr., Marmara Üniversitesi Teknik Eğitim Fakültesi

Government is obligated to assist growth of the state enterprises. Government has to act as an umpire and specify the required rules. (1)

Growth is a particularly important concept for developing countries such as ours. In parallel to population growth, people's needs increase and therefore production has to be increased. Furthermore, efficient use of economic resources is the most significant aspect of economic growth. Economic growth is very important in production management. As a result of increased economic growth, businesses experience development in line with growth in production, productivity and profitability targets. From the macro point of view, economic growth leads to social welfare in countries. (2)

As a result of the aforesaid, health, defense, transportation, education etc sectors develop. Greatest underlying factor that makes countries and businesses take an interest on growth is that it allows seeing the future, and it is very important to plan the steps that will be taken for the future. Forecasting the growth rate is one of the essential parameters of the planning process. Even though there are many parameters that impact the growth rate; in this study import, export, production, index increase, employment, population and foreign exchange rate have been taken into consideration.

ANN models have been applied to a large number of problems because of their non-linear system modeling capability by learning ability using collected data. They offer highly parallel, adaptive models that can be trained by experience.(4) In fact, ANN models have the universal approximation property that means under mild conditions on the data, they can fit any data set with an arbitrary high precision, provided that there are a sufficient number of parameters in the model. (5)

The use of ANNs for forecasting received great attention from many different fields. A wide range of business forecasting problems have been solved by neural networks. Some of these application areas include accounting (forecasting accounting earnings, earnings surprises; predicting bankruptcy and business failure), finance (forecasting stock market movement, indices, return, and risk; exchange rate; futures trading; commodity and option price; mutual fund assets and performance), marketing (forecasting consumer choice, market share, marketing category, and marketing trends), economics (forecasting business cycles, recessions, consumer expenditures, GDP growth, inflation, total industrial production, and US Treasury bond), production and operations, international business (predicting joint venture performance, foreign exchange rate), real estate (forecasting residential construction demand, housing value).(3)

Since the reliability of data is very important for forecasting, the data obtained by the State Statistics Institute after 1985 to 2005 are used here. From 1985 to 2002 years data has been used to learning aim and from 2003 to 2005 years data has been used to testing aim. Future growth rates which include years 2003, 2004, 2005 have been forecasted of the above mentioned parameters via time series analysis and ANN, and the results were compared.

2. MODELLING BY ANN

There are multitudes of different types of ANN models. Some of the more popular of them include the multilayer perceptron, which is generally trained with the backpropagation algorithm.

2.1. Perceptron

Early ANN models, consisting of a single layer and simple threshold functions, were called perceptrons. The architecture of a perceptron is shown in figure 1.

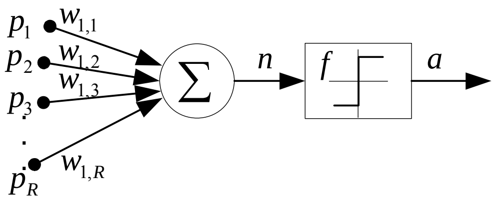

Fig. 1 The architecture of a single layer perceptron

The outputs of a perceptron are calculated by summing the weighted inputs coming from its input links, so that

$$\mathrm{n} = \mathrm{Wp} + \mathrm{b} \tag{1}$$

$$\mathrm{a} = \mathrm{f(n)} \tag{2}$$

where in the multiple neuron case n is the sum of the weighted input vector, **W** is a SxR weighting coefficients matrices, R is the input number, S is the neuron number, **p** is a Rx1 input vector, b is the bias vector, $\mathrm{f(.)}$ f(.) is the activation function, and **a** is output vector. In perceptrons, the activation function of output layer is hard limiter:

$$a = \begin{cases} 0 & , f(n) < 0 \\ 1 & , f(n) \geq 0 \end{cases} \tag{3}$$

Perceptrons must be implemented by a training rule for adjusting the weighting coefficients. In the training process, it compares the actual network outputs to the desired network outputs to determine the new weighting coefficients,

$$\mathbf{e} = \mathbf{t} - \mathbf{a} \tag{4}$$

$$\mathbf{W}^{new} = \mathbf{W}^{old} + \mathbf{ep}^{\mathrm{T}} \tag{5}$$

$$\mathbf{b}^{new} = \mathbf{b}^{old} + \mathbf{e} \tag{6}$$

where **e** is the vector errors between the actual and desired outputs, **t** is the desired (target) vector.

2.2. Backpropagation

Backpropagation is a training method for multilayer feedforward networks. Such a network including M-layers of perceptrons with bias adjustments is shown in figure 2.

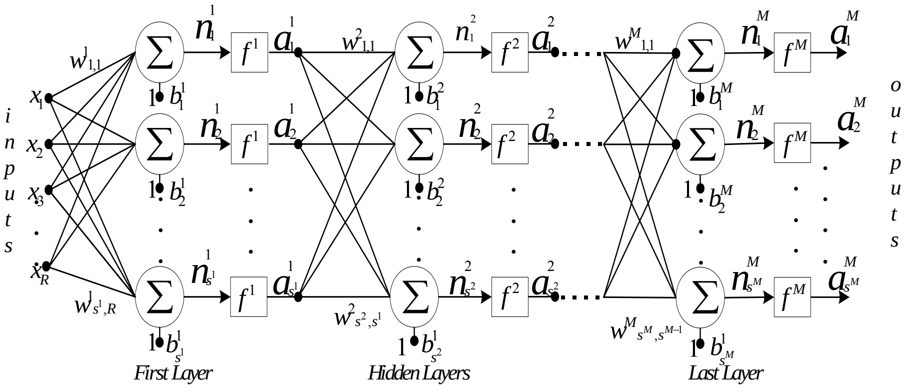

Fig. 2. M-layer feedforward network

Training process is initiated from the output layer. Then, the error calculated for output neurons is propagated to the backward through the weighting factors of the network. The state of the network always changes in such a way that the output follows the error curve of the network towards down, so that the error decreases repeatedly. This algorithmic approach is known as gradient descent algorithm. (6) First step in training is propagating the inputs towards the forward layers through the network:

$$\begin{aligned} &\mathrm{a}^0 = \mathrm{p} \\ &\mathrm{a}^{\mathrm{m}} = \mathrm{f}^{\mathrm{m+1}}(\mathrm{W}^{\mathrm{m+1}}\mathrm{a}^{\mathrm{m}} + \mathrm{b}^{\mathrm{m+1}}) \quad \textit{for} \quad m = 0,2,\ldots,M-1, \\ &\mathrm{a} = \mathrm{a}^{\mathrm{M}} \end{aligned} \tag{7}$$

where *m* is the layer index, *M* is the last layer number in the network. In Eq.6 f^{m} and W^{m} represent the activation function vector and weighting coefficients matrices related to the layer *m*, respectively.

Second step is propagating the sensivities (**s**) from the last layer to the first layer through the network: $\mathrm{s}^{\mathrm{m}}, \mathrm{s}^{\mathrm{m-1}}, .., \mathrm{s}^2, \mathrm{s}^1$. It can be expressed in matrix form as:

$$\begin{aligned} &\mathbf{s}^{\mathbf{M}} = -2\dot{\mathbf{F}}^{\mathbf{M}}(\mathbf{n}^{\mathbf{M}})\ (\mathbf{t}-\mathbf{a}) \\ &\mathbf{s}^{\mathbf{m}} = \dot{\mathbf{F}}^{\mathbf{m}}(\mathbf{n}^{\mathbf{m}})\ (\mathbf{W}^{\mathbf{m+1}})^{\mathbf{T}}\ \mathbf{s}^{\mathbf{m+1}} \quad \boldsymbol{for} \quad \boldsymbol{m = M-1,..,2,1} \end{aligned} \tag{8}$$

Where;

$$\dot{\mathrm{F}}^{m}(\mathrm{n}^{m}) = \begin{bmatrix} \dfrac{\partial f^{m}(n_1^m)}{\partial n_1^m} & 0 & \Lambda & 0 \\ 0 & \dfrac{\partial f^{m}(n_2^m)}{\partial n_2^m} & \Lambda & 0 \\ \mathrm{M} & \mathrm{M} & & \mathrm{M} \\ 0 & 0 & \Lambda & \dfrac{\partial f^{M}(n_{s^M}^M)}{\partial n_{s^M}^M} \end{bmatrix} \tag{9}$$

The last step is updating the weights and biases using approximate steepest descent rule:

$$\begin{aligned} \mathbf{W}^{\mathrm{m}}(\mathrm{k}+1) &= \mathbf{W}^{\mathrm{m}}(\mathrm{k}) - \alpha \mathbf{s}^{m} (\mathbf{a}^{\mathrm{m}-1})^{\mathrm{T}} \\ \mathbf{b}^{\mathrm{m}}(\mathrm{k}+1) &= \mathbf{b}^{\mathrm{m}}(\mathrm{k}) - \alpha \mathbf{s}^{m} \end{aligned} \tag{10}$$

where α represents the training rate, and k represents the epoch number.

2.3. Training

ANN must be trained before it becomes useful. Some ANN models employ supervised training while others are referred to as unsupervised or self-organizing training. The training set consists of presenting input and output data to the network.

Vast majority of ANN models are supervisory type networks. In the training phase, the actual output of ANN is compared with the desired output. Then the network adjusts the weighting coefficients, which usually begin with random set, so that the next iteration will produce a closer match between the desired and the actual output. The training method tries to minimize the current errors for all processing elements.

This global error reduction is created over time by continuously modifying the weighting coefficients until acceptable network accuracy is reached. The training continues until the ANN reaches user defined performance level.

Training phase may take a lot of time. This level signifies that the network has achieved the desired statistical accuracy for a given sequence of inputs. When no further training is necessary, the weighting coefficients are frozen for the application. After a supervised network performs well on the training data, then it is important to see what it can do with data it has not seen before. If a system does not give reasonable outputs for this test set, the training period is not over. Indeed, this testing is critical to insure that the network did not simply memorize a given set of data, but learned the general patterns involved within an application.

3. MODELLING BY TIME SERIES

Time series data are a series of observations recorded in sequence over time. Interval between the observations is determined depending on the type of time series data. Usual intervals are hours, days, weeks, months, quarters, or years. In most cases, it is unreasonable to assume that time series data are independent and identically distributed. As a matter of fact, it is essential to allow for statistical dependence over time. With time series data, the annual growth rate can be learned and therefore past values of variables can be to predict the future values.

For time series regression and prediction, let Y_t denote an observation in date t for t = 1; ..; T (denotes the total number of observations). In time series analysis, one natural question is whether and what extent we can predict the future values of a time series variable using past values. One popular model for answering this question is called auto regression (AR).

An AR model is a regression model that relates a time series variable to its past values. More specifically, the p-th order AR model (AR(p)) has the form:

$$\boldsymbol{Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \ldots + \beta_p Y_{t-p} + U_t} \qquad (11)$$

where the regression coefficients $\beta_0, \beta_1, \ldots, \beta_p$ are unknown, and U_t is unobserved error term that satisfies $E(U_t \mid Y_{t-1}, Y_{t-2}, \ldots..) = 0$.

One implication of this assumption is that U_t is uncorrelated with any past values Y_{t-k} for any k. The model assumes that only the most recent p lags enter the regression AR(p) model. The number of lags p is often called the lag length of the AR. AR(p) models can be estimated using the ordinary least squares (OLS) estimator. To describe the OLS estimator for the AR(p) model we need some notation. Let **b** = (b_0, b_1,....,b_p), **X** be the (T-p)*(p+1) matrix such that the t-th row is (1, Y_{t-1},… ,Y_{t-p}), and $\boldsymbol{Y}$ be the (T-p)*1 vector such that the t-th element is Y_t for $t = p$+1,….,T. The OLS estimator of b is;

$$\boldsymbol{\hat{\beta} = (X^T X)^{-1} X^T Y} \qquad (12)$$

It can be shown that the OLS estimator is asymptotically consistent for β. That is, β converges to $\boldsymbol{\hat{\beta}}$ in probability as T goes to infinity. The practical implication of this is that the OLS estimator will be close to the true unknown regression coefficients when T is quite large.

4. PREDICTION WITH AN ANN MODEL

The prediction of growth rate is a vital importance for economy management, production planning and budget planning. Annual growth rate has been predicted by an autoregressive (AR) model and ANN. The data obtained by the State Statistics Institute after 1985 is used here. (7) (Table 1)

Even though there are many parameters that impact the growth rate; import, export, production, index increase, employment, population and foreign exchange rate; in this study have been taken into consideration.

Table 1. Used data which obtained by the State Statistics Institute from 1985 to 2005

Years	Export	Import	Growth Rates	Employ-Ment	Popu-lation	Total Industry Index	Dollar Rate
1985	7958009,70	11343376,35	1,70	17227,00	50307,00	7400000,00	451,40
1986	7456725,60	11104771,29	4,40	17532,00	51433,00	7740000,00	581,05
1987	10190049,42	14157806,91	7,50	17921,00	52561,00	8800000,00	753,65
1988	11662024,12	14335397,81	-0,70	17668,00	53715,00	8510000,00	1117,65
1989	11624691,72	15792142,91	-0,60	17997,00	54894,00	8820000,00	1881,19
1990	12959287,61	22302125,59	6,80	18681,00	56098,00	9520000,00	2347,30
1991	13593462,02	21047013,87	-1,58	18171,00	57193,00	9950000,00	3041,90
1992	14714628,83	22871055,11	4,41	18462,00	58248,00	10610000,00	6486,01
1993	15345066,89	29428369,53	6,16	18806,00	59323,00	12140000,00	8814,34
1994	18105872,08	23270019,03	-7,77	22316,00	60417,00	10740000,00	17203,52
1995	21637040,88	35709010,77	6,08	20833,00	61532,00	12430000,00	40393,00
1996	23224464,97	43626642,50	5,32	21539,00	62667,00	12410000,00	62503,00
1997	26261071,79	48558720,67	8,67	21008,00	63823,00	14400000,00	116000,00
1998	26973951,74	45921392,21	2,25	21374,00	65001,00	13120000,00	214190,00
1999	26587224,96	40671272,03	-7,38	21860,00	66200,00	13750000,00	330605,00
2000	27774906,05	54502820,50	1,40	20500,00	67421,00	12750000,00	557782,00
2001	31334216,36	41399082,95	-11,10	20704,00	68618,00	12410000,00	679162,00
2002	36059089,03	51553797,33	6,37	21658,00	69626,00	14400000,00	1320839,00
2003	47252836,30	69339692,06	4,24	20811,00	70712,00	13120000,00	1642218,00
2004	63167153,00	97539765,97	8,21	21563,00	71789,00	13750000,00	1337001,00
2005	73476408,00	116774151,00	7,20	22838,00	72065,00	12750000,00	1328700,00

(Annual of Turkey Statistics, 2006)

Neural networks are data-driven techniques. Therefore, data preparation is a critical step in building a successful neural network model. Without a good, adequate, and representative data set, it is impossible to develop a useful, predictive ANN model. Thus, the reliability of ANN models depends to a large extent on the quality of data. There are several practical issues around the data requirement for an ANN model. The first one of these is the size of the sample used to build a neural network. While there is no specific rule that can be followed for all situations, advantage of having large samples should be clear because not only do neural networks have typically a large number of parameters to estimate, but also it is often necessary to split data into several portions to avoid over fitting, select model, and perform model evaluation and comparison. Of course, if data in the sample are not homogeneous or the underlying data generating process in a time series changes over time, then a larger sample may even hurt performance of static neural networks as well as other traditional methods.

4.1. Normalizing the data

Normalization of data is a process of scaling the numbers in a data set to improve the accuracy of the subsequent numeric computations and is an important stage for training of the ANN. Normalization also helps in shaping the activation function. For this reason, the normalization function has been used.

$$\mathbf{X_{norm} = (x - X_{min}) / (x_{max} - x_{min})} \quad (13)$$

4.2. Selecting the ANN architecture

Many types of ANN have been used for forecasting. However, the multilayer feedforward architecture is by far the best developed and most widely applied one for forecasting applications. Therefore, our discussion will be focused on this type of neural network, although it may be applied to other types of ANN as well.

The number of layers and the number of processing elements per layer are important decisions for selecting the ANN architecture. Choosing these parameters to a feedforward, back-propagation topology is the art of the ANN designer. There is no quantifiable, best answer to the layout of the network for any particular application. There are only general rules picked up over time and followed by most researchers and engineers applying this architecture to their problems. The first rule states that if the complexity in the relationship between the input data and the desired output increases, then the number of the processing elements in the hidden layer should also increase.

The second rule says that if the process being modeled is separable into multiple stages, then additional hidden layer(s) may be required. In practice, in most of the forecasting applications only one hidden layer and some small hidden nodes are used.

For this reason, some tests are applied to choose the number of neurons for the first and second layers. Neuron numbers of the hidden layer are selected 5,10,15 and 20 respectively and according to these numbers 4 different learning processes are realized. Obtained results show that 15 neuron is the best among neuron numbers. These results are given in table 2.

For forecasting applications, the most popular transfer function for hidden nodes is either logistic or hyperbolic and it is the linear or identity function for output nodes. If the data, especially the output data, have been normalized into the range of [0, 1], then sigmoid function can be used. In this application, learning rate (α) is selected 0.5 in all experiments and activation function is determined as sigmoid in input and hidden layers, as tangent in output layers.

4.3. Training the network

Once a network has been structured for a particular application, that network is ready to be trained. To start this process the initial weights are chosen randomly. Then, the training begins. The network then processes the inputs and compares its result outputs against the desired outputs as given in Eq.7. Errors are then propagated back through the system, causing the system to adjust the weights, which controls the network. This process occurs over and over as the weights are continually decreased as given in Eq.10. The set of data, which enables the training, is called the training set. During the training of a network the same set of data is processed many times as the connection weights are ever refined.

ANN Simulator has been trained for a thousand epochs in this network type and parameters, when the error becomes stable the training is stopped. (Fig.2)

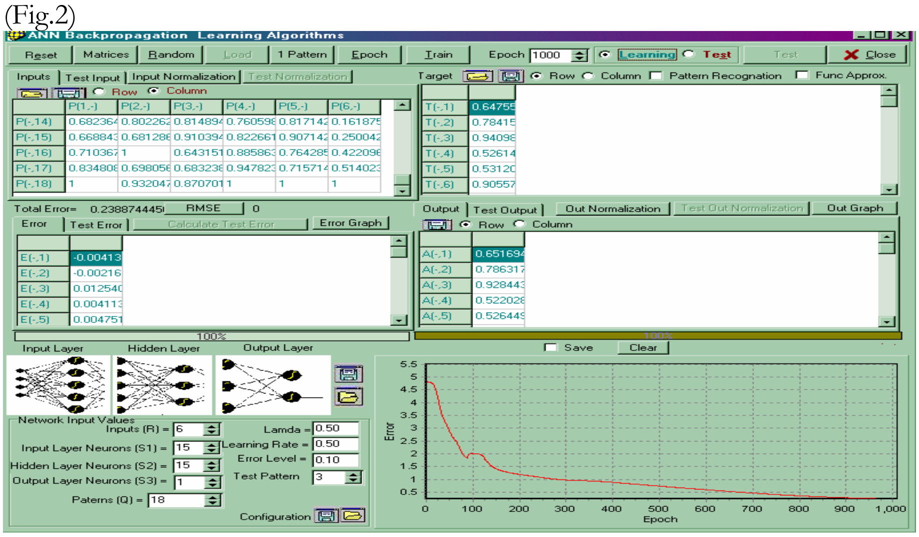

Fig.2. ANN simulator screen on the learning task

Neuron numbers of hidden layer are selected 5, 10, 15 and 20 respectively. End of the training task calculated error and regression coefficient are given in table 2. Also, for the four different neural models, founded normalized growth rate values with the real growth rate values are drawn in Fig 3.

Table2. Training results with different neuron numbers of hidden layer

Model	Normalized Total Error	Normalized RMS Error	Regression Coefficient(R^2)
ANN 5	0,922	0,070	0.936980
ANN 10	0,607	0,051	0.966451
ANN 15	0,169	0,032	0.996986
ANN 20	0,151	0,031	0.999216

Fig. 3. The trained data with different neuron numbers

4.4. Testing the network

It is important to note that the test sample served as out-of-sample should not in anyway be used in the model-building process. If the cross-validation is used for model selection and experimentation, the performance on the validation sample should not be treated as the true performance of the model. Although some of the above issues are unique to neural networks, some are general issues to any forecasting method. In the test, the ANN is presented with the unknown input patterns and the output is calculated.

Future growth rates between years 2003 and 2005 has been forecasted with the above mentioned parameters via ANN. End of the testing task for the four different neural models calculated error values are given in table 3, normalized growth rate values with the real growth rate values are drawn in Fig 4.

Table 3. Test results with different neuron numbers of hidden layer

Model	Normalized Total Error	Normalized RMS Error	Regression Coefficient(R^2)
ANN 5	0,23	0,08	0.069902
ANN 10	0,22	0,07	0.514133
ANN 15	0,18	0,06	0.186236
ANN 20	0,11	0,04	0.995920

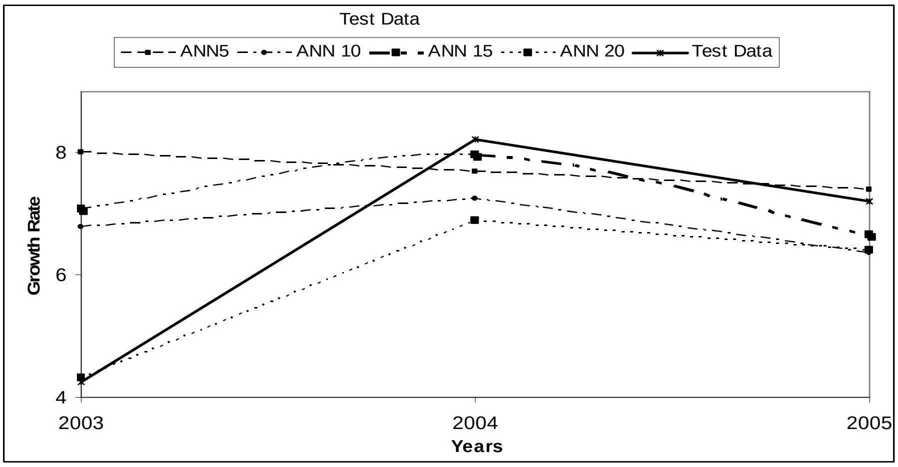

Fig. 4. Test result of ANN.

5. PREDICTION BY AR MODEL

For the comparison, AR model has been used. After the search of the best fit with E-wievs Package Programme, AR model has been found as below equation.

GROWTH RATE = - 1.654405116 e-006* EXPORT + 0.0008924357788 * EMPLOYMENT + 1.146814125e-006* İMPORT + 2.763129112e-006* RATE - 0.002427582452* POPULATİON + 2.969663428e-006* TOTAL İNDUSTRY INDEX+ 91.16326284 (13)

According to equation (13) regression graph is drawn in fig 5.

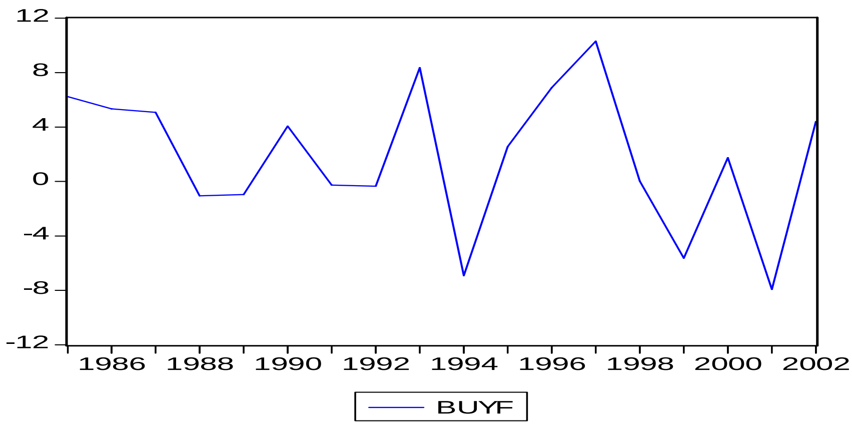

Fig. 5. Regression graph of AR Model

Future growth rates which include the years 2003, 2004, 2005 have been forecasted of the above mentioned parameters via AR Model. From 2003 to 2005 years data used to testing aim and drawn graphically. These results are given in Fig 6.

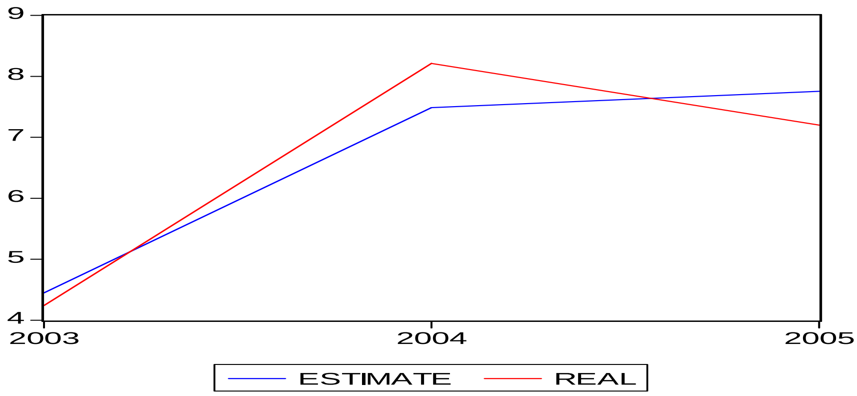

Fig. 6. Test result of AR Model

Aim of Forecasting, regression coefficient (R^2) calculated as 0.811476 and Durbin-Watson value calculated as 2.390806. This results show that there is a strong relation between dependent and independent variables.

6. CONCLUSION

Artificial neural networks have emerged as an important tool for business forecasting. ANNs have many desired features that are quite suitable for practical forecasting applications. In this paper, annually growth rate has been predicted by AR model and an ANN model. Since the reliability of the data is very important for forecasting, the data obtained by the State Statistics Institute after from 1985 to 2005 are used here. From 1985 to 2002 years the data was used for learning purposes and from 2003 to 2005 years it was used to testing purposes. Future growth rates which include 2003, 2004, 2005 years have been forecasted of the above mentioned parameters via time series analysis and ANN, and the results were compared**.** The results have shown that the prediction error obtained by ANN model is smaller than the error obtained by AR model. When neuron numbers of input layer are increased, prediction error obtained by ANN is decreased.

REFERENCES

Iraz Rıfat ; "'*Küresel Rekabet Ortamında Küçük ve Orta Boy İşletmelerin Ulusal, Sosyal-Ekonomik Sisteme Katkıları Açısından Değerlendirilmes*i", www.sosyalbil.selcuk.edu.tr., Erişim tarihi. 08.05.2007.

Tekin Mahmut, Ömürbek Nuri ; "*Küresel Rekabet Ortamında Teknolojik İşbirliği*", Nobel yayınevi, Ankara, 2004, S.103

Zhang, G. Peter; "*Neural Networks in Business Forecasting",* Hershey, PA, USA: Idea Group Inc., 2003. p 15.

DIA H., "*An Object-oriented Neural Network Approach to Short-term Traffic Forecasting*", European Journal of Operational Research 131, Elsevier, 2001, 253-261.

Hagan, T. M, Demuth, H.B, Beale, M.; "*Neural Network Design*", PWS Publishing Company, 1996, 2-44.

Tektas, M., Topuz, V.,Tektas,N.; "*Artificial Neural Networks Education Simulator*", IMS' 2004, 4th International Symposium on Intelligent Manufacturing Systems, September 2004 Sakarya, Turkey, 1196-1206

www.ekodiyalog.com, Erişim tarihi. 08.05.2007