Available online at www.alphanumericjournal.com

alphanumeric journal

The Journal of Operations Research, Statistics, Econometrics and Management Information Systems

Volume 9, Issue 2, 2021

Received: July 15, 2021
Accepted: October 27, 2021
Published Online: December 31, 2021

AJ ID: 2021.09.02.STAT.02
DOI: 10.17093/alphanumeric.972138
Research Article

# Artificial Neural Network approach on Type II Regression Analysis

Berkalp, Tunca

Afyonkarahisar, Turkey, berkalp0606@gmail.com

Sinan Saraçlı, Ph.D. *

Assoc. Prof., Department of Statistics, Faculty of Science and Literature, Afyon Kocatepe University, Afyonkarahisar, Turkey, ssaracli@aku.edu.tr

* Ahmet Necdet Sezer Kampusü, Gazlıgöl Yolu, 03200, Afyonkarahisar, Turkey

**ABSTRACT**

In this study, the Artificial Neural Network (ANN) approach was applied to the OLS-Bisector technique, which is one of the Type II Regression techniques, through this study. In order to measure the performance of this newly created ANN-Bisector technique, it was compared with the OLS-Bisector technique. First of all, literature information on ANN and OLS-Bisector Regression techniques is given, and the features of two techniques are mentioned. In line with this information, a comparison was made between OLS based bisector technique and ANN based bisector techniques. In order to compare these two techniques, they were modeled in different distributions and in different sample sizes. In order to compare the performances of these models, the "Mean Absolute Percent Error" (MAPE) criterion was used. As a result of the study, it was seen that the ANN based bisector technique gave better results with lower error than the OLS based bisector technique. With this study, it is foreseen that it will represent an example for researchers who want to work in these fields in the future.

**Keywords:** Artificial Neural Network, Type II Regression, Measurement Error Models

# 1. Introduction

In general, neural networks are computer programs that mimic the neural networks of the human brain. Artificial neural networks can in a sense be considered as a parallel information processing system. This information is given to the artificial neural networks by educating the samples on the related event. In this way, various generalizations can be made on the properties that are revealed by means of examples, and then they can also be produced for the events that have not yet been encountered (Aygören vd., 2012).

It can be said that the computational and information processing power of ANN comes from its parallel distributed structure, ability to learn and generalize. Generalization is defined as the ANN producing appropriate responses for inputs that are not encountered in the training or learning process. These superior features show the ability of ANN to solve complex problems. Today, in many fields of science, ANN has been effective and has found application due to the following features (Ergezer et al., 2003):

- Nonlinearity: The cell, which is the basic processing element of ANN, is not linear. Therefore, ANN, which is formed by combining cells, is not linear and this feature is spread over the whole network. With this feature, ANN has become an important tool in solving complex nonlinear problems.
- Learning: In order for the ANN to display the desired behavior, it must be adjusted in accordance with the purpose. This means that the correct connections must be made between cells and that the connections must have appropriate weights. Due to the complex structure of ANN, links and weights cannot be given or designed as preset. For this reason, ANN should learn the problem by using the training examples it receives from the problem it is interested in so that it shows the desired behavior.
- Generalization: After learning the problem it is interested in, ANN can also produce the desired response for test samples that it has not encountered during training. For example, for character recognition purposes, an ANN can give correct characters for corrupted character inputs, or a trained ANN model of a system may behave the same as the system for input signals not given during the training process.
- Adaptability: ANN adjusts its weights according to the changes in the problems it is interested in. That is, the ANN, which is trained to solve a specific problem, can be retrained according to the changes in the problem, and if the changes are continuous, the training can be continued in real time. With this feature, ANN is effectively used in areas such as adaptive sample recognition, signal processing, system diagnosis and control.
- Error tolerance: ANN has a parallel distributed structure because it consists of many cells connected in various ways, and the information of the network is distributed over all connections in the network. Therefore, the inactivation of some links or even some cells of a trained ANN does not significantly affect the network's ability to produce accurate information. Therefore, their ability to tolerate error is extremely high compared to traditional methods.

The most important reason why ANN is widely used today is its ability to learn. ANN has been developed to perform the skills of the human brain such as deriving new information, creating new information and discovering it through learning without assistance. In addition to learning, he also has the ability to make connections between information. Some usage areas and basic functions of ANN are stated as follows (Uğur and Kınacı, 2006):

- Prediction or forecasting: future sales, weather forecasts, horse races, environmental risks,
- Classification and Clustering: Customer profiles, medical diagnostics, voice and shape recognition, cell types,
- Control (Control): Sound and vibration levels in aircraft for early warning,

In studies, it is known that classical regression techniques are always the first to come to mind and the most used techniques in investigating the compatibility between two methods. When the measurements made with both techniques are examined, it is seen that the independent variable does not contain any measurement error compared to classical regression techniques when it is accepted as dependent and independent variable. However, the measurement made with these two techniques actually contains some error. Based on this idea, the models established by considering that both observation variables contain errors are called Type II regression models in the literature (Saraçlı, 2008).

Type II regression models are divided into two groups as linear and nonlinear regression models. Some of the Type II regression techniques are as follows; OLS (X | Y), OLS (Y | X), Theil, OLS, Theil, Orthogonal (OR, Major Axis, MA), Reduced Major Axis (RMA), Deming, Optimal-Deming, Passing-Bablok, York and Optimal York Regression Technique (Saraçlı, 2008). In this study, OLS-Bisector Regression technique used on Type II Regression techniques. Therefore only information will be given about the OLS-Bisector technique. For more information on Type II regression techniques, the source Tunca (2019) can be examined.

The OLS Bisector technique tries to minimize the distance of the observation points from the predicted regression line by considering the bisector of the OLS(Y|X) line and the OLS(X|Y) line. Although the inversion of the OLS(X|Y) line has caused a lot of controversy, no studies have been found about the lack of the OLS Bisector line (Isobe et al., 1990; Saraçlı 2011).

## 2. Methods

### 2.1. Artificial Neural Networks

Artificial neurons are also called Node, Unit or Processor Element. Layers are formed when neurons come together in the same direction. There are three types of layers in an artificial neural network. Input Layer, Intermediate/Hidden Layer and Output Layer. Networks are formed by arranging the layers in a certain order. Networks are named according to the number of layers, their order and learning algorithms (Bayır, 2006). Recurrent networks, such as Hopfield nets (Hopfield, 1982, 1984), allow feedback between layers. Figure 1 illustrates a three-layer network:

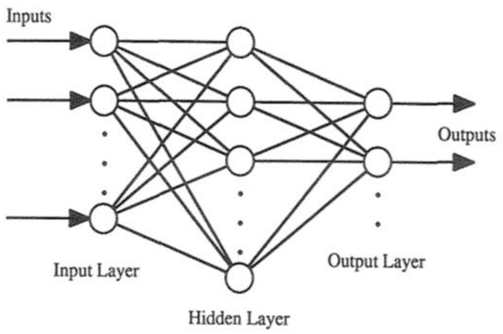


**Figure 1.** A basic overview of artificial neural network topology.

There is no fixed idea about how many nodes should be included in the hidden layer in ANN. If there are too many nodes when building a neural network, the network may not be able to generalize against different problems that it does not encountered before. On the other hand, if there are too many nodes in the hidden layer, it may take an unacceptable or long time for the network to learn anything of any value (Dawson and Wilby, 1998).

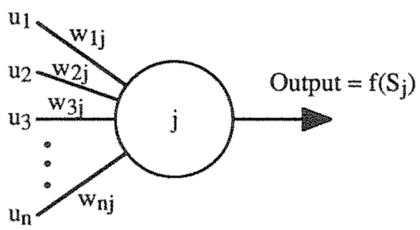


**Figure 2.** An artificial neuron

On closer inspection of a single neuron in Figure 2, it has a range of input information for each j neuron, from $u_1$, to $u_n$. Here, each of the input information entering the network is multiplied by the $w_{ij}$ weights and collected in the j neuron. It is calculated as in equation (1):

$$S_j = \sum_{i=1}^{n} w_{ij} u_j + w_{0j} \tag{1}$$

In Equation (1), $w_{0j}$, called bias, is included. An activation function is applied to the $S_j$ value to calculate the final output from the neuron. This activation function can be linear, discrete, or any continuous distribution function. Some activation functions used in the literature: step function, linear function, sigmoid function etc. For an activation function to be a good function, it must be differentiable, nonlinear, monotonically increasing or decreasing. For example, when it comes to a learning algorithm such as back propagation, the sigmoid function provides the appropriate criteria for this, and it is a function generally used in feedforward neural networks when the literature is examined (Dawson and Wilby, 1998). This function is given in Equation (2).

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

**Back-Propagation Algorithm (BP)**

Back propagation algorithm is a supervised learning algorithm. For certain input model ($U^p$), its output in the network ($Y_i^2$) must match its target output ($t_i$). This algorithm, which aims to update the appropriate weights in this way, consists of three phase. These is forward phase, error back-propagation phase, weight update phase. These phase explain are given below (Suresh et al, 2005).

*Forward phase*: For a given input pattern $U^0$, the activation values of hidden and output layer are computed as follows:

$$Y_i^l = f(\sum_{j=1}^{N^{l-1}} W_{ij}^l U_j^{l-1}),\ l = 1,2,\ldots,N^l \tag{3}$$

where $f(.)$ is a bipolar sigmoid function. Let the time taken to calculate the multiplication, addition, multiplication and activation value, respectively, be $t_a$, $t_m$ and $t_{ac}$. The time taken to complete the forward phase is calculated as in Equation (4):

$$t_1 = N^1(N^0 + N^2)M_a + t_{ac}(N^2 + N^1) \tag{4}$$

where $M_a = t_m + t_a$.

***Error back-propagation phase:***

At this phase, the bias value determined between the output of the neural network and the target value is propagated back to all neurons through the output weights, and the value of $\zeta_i^2$ for the neurons in the output layer is calculated as in equation (5).

$$\zeta_i^2 = \left(Y_i^2 - t_i\right)f'(.)\ i = 1,2,\ldots,N^2, \tag{5}$$

where $f'(.)$ is the first derivative of the activation function. The term ($\zeta_i^1$) for ith hidden neuron is given by:

$$\zeta_i^1 = f'(.)\sum_{j=1}^{N^2} W_{ji}^2 \zeta_i^2\ \ i = 1,2,\ldots,N^1, \tag{6}$$

The time taken to complete the error back-propagation phase is represented by $t_2$ and is calculated as:

$$t_2 = N^1 N^2 M_a. \tag{7}$$

*Weight update phase*: In this phase, the network weights are updated and the updation process of any $W_{ij}^l$ depends on the value of $\zeta_j^l$ and $Y_i^{l-1}$.

$$W_{ij}^l = W_{ij}^l + \eta\zeta_i^1 Y_i^{l-1}\ j = 1,2, \tag{8}$$

where $\eta$ is the learning rate. The time taken to update the weight matrix between the three layers is represented by $t_3$ and it is equal to

$$t_3 = N^1(N^2 + N^0)M_a. \tag{9}$$

Let $t_{ac} = \beta M_a$. The total processing time ($T_{seq}$) for training a single pattern is the sum of the time taken to process the three phases and is given as

$$\begin{aligned} T_{seq} &= t_1 + t_2 + t_3 \\ &= [N^1 K + \beta N^2]M_a, \end{aligned} \tag{10}$$

where $K = 2N^0 + 3N^2 + \beta$.

**Training Method**

There are many training methods in artificial neural networks. The method considered the fastest of these is the Levenberg-Marquardt method. Levenberg-Marquardt method was used in this study. The Levenberg-Marquardt algorithm (LM) is an approximation to the Newton method (Figure 4) used also for training ANNs (Khan and Sahai, 2012).

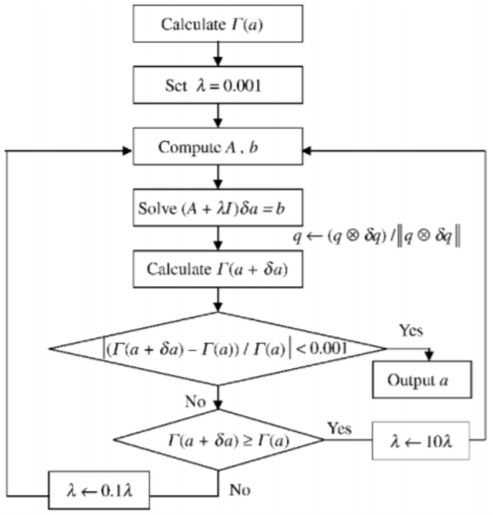


**Figure 4.** Flowchart of Levenberg-Marquardt

The Newton method approximates the error of the network with a second order expression, which contrasts to the Backpropagation algorithm that does it with a first order expression. LM updates the ANN weights as follows (Khan and Sahai, 2012):

$$\Delta \mathrm{w} = \left[ \mu I + \sum_{p=1}^{p} J^{p}(w)^{T} J^{p}(W) \right]^{-1} \nabla E(w) \qquad (11)$$

where, $J^p$ (w) is the Jacobian matrix of $e^p$ (w), an error vector evaluated in *w*, and *I* is the unit matrix. For pattern p, vector error $e^p$ (w) is network error, so $e^p$ (w) = $t^p$ −$o^p$ (w). The μ parameter is increased or decreased at each step. If the error is reduced, μ is multiplied by a factor β and otherwise multiplied by β. The LM performs the Jacobian matrix, network output, and error vectors for each model, following the steps detailed in Figure (4). Then, the Δw value is calculated as in Equation (11) and recalculates the calculated error as w+Δw and the weights of the neural network. If the error is reduced, the μ value is divided by β, keeping the new weights and this process starts again, but if the error has not decreased, the μ value is multiplied by β, Δw is calculated again with a new value (Khan and Sahai, 2012). The formulaic graphic representation of the LM algorithm is given in Figure 5:

Initialize Weights;

While not Stop-Criterion do

  Calculates cp (w) for each pattern

  $e_1 = \sum_{p=1}^{p} e^p(w)^T e^p(w)$

  Calculates Jp (w) for each pattern

  Repeat

    Calculates $\Delta$w

    $e_2 = \sum_{p=1}^{p} e^p(w+\Delta w)^T e^p(w+\Delta w)$

    If e1 ≤ e2 Then

      $\mu = \mu * \beta$

    end If

  Until e1 < e2

  $\mu = \mu/\beta$

  w = w + $\Delta$w

End While

**Figure 5.** Pseudocode of Levenberg-Marquardt algorithm

### Transfer Function

When the transfer functions for artificial neural network are examined, there are quite a lot of transfer functions. In the literature, researchers generally used logsig and tansig functions. As a result of the experiments, the logsig function was used because the logsig function gave better results than the tansig function.

$$logsig(n) = \frac{1}{(1+\exp(-n))}$$

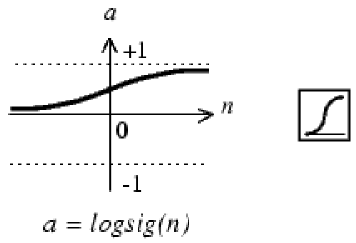


**Figure 6.** Graph for logsig function (MATLAB User's Guide, 2020)

## 2.2. Type II Regression

The Least Squares (Least Squares) approach, which is widely used in classical regression, assumes that the source of the error term is only from the dependent variable. Regression techniques that assume that the independent variables in the model may contain measurement error can be called Type II regression techniques (Gazeloğlu and Saraçlı, 2013).

There are many different Type II regression techniques in the literature. (Saraçlı et al., 2009). In general, these techniques, which are based on the logic of taking into account the errors in both variables as a result of taking the distances of the observation values that are perpendicular to the obtained regression equation or calculated depending on the amount of error, are Orthogonal Regression, Deming Regression, York Regression techniques and their derivations under various conditions. Orthogonal Regression Technique in terms of their calculations in estimating the regression parameters; Deming Regression Technique; Deming is divided into three groups as Optimal Deming and Weighted Deming, while York Regression Technique is divided into two groups as York and Optimal York Regression Techniques. The Passing-Bablok Regression Technique is another non-parametric regression technique that is an alternative to the EKK Technique (Saraçlı, 2011). In the case of a real dataset with an unknown distribution, if both dependent and independent variables contain measurement errors, using Deming or EKK-Anchorage techniques can provide more reliable results (Saraçlı and Türkan, 2012).

In this study we examined the performance of ANN for Type II regression analysis. Because of the results of earlier Type II regression studies showed that the Bisector technique gives the better results, here we calculate two regression lines, considering X and Y variables independent respectively and then calculated the bisector regression line using OLS and ANN approaches (Tunca, 2019).

The OLS Bisector regression technique can be defined as a line that mathematically divides the OLS(Y|X) and OLS(X|Y) regression lines into two (ie, takes the bisector) (Saylor et al., 2006).

The slope coefficient of the OLS Bisector line is calculated with the help of the formula in Equation (12). In addition, the variance of this line is calculated with the help of Equation (13), and its covariance is calculated with the help of Equation (14).

$$\hat{\beta}_{Bis} = \left(\hat{\beta}_1 + \hat{\beta}_2\right)^{-1}\left[\hat{\beta}_1\hat{\beta}_2 - 1 + \sqrt{\left(1+\hat{\beta}_1^2\right)\left(1+\hat{\beta}_2^2\right)}\right] \tag{12}$$

Here $\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}}$ is the slope of OLS(X|Y) regression and $\hat{\beta}_2 = \frac{S_{yy}}{S_{xy}}$ is the slope of OLS(Y|X) regression.

$$Var\left(\hat{\beta}_{Bis}\right) = \frac{\hat{\beta}_{Bis}^2}{\left(\hat{\beta}_1+\hat{\beta}_2\right)^2\left(1+\hat{\beta}_1^2\right)\left(1+\hat{\beta}_2^2\right)}\left[\left(1+\hat{\beta}_2^2\right)^2 Var\left(\hat{\beta}_1\right) + 2\left(1+\hat{\beta}_1^2\right)\left(1+\hat{\beta}_2^2\right)Cov\left(\hat{\beta}_1,\hat{\beta}_2\right) + \left(1+\hat{\beta}_1^2\right)^2 Var\left(\hat{\beta}_2\right)\right] \tag{13}$$

and

$$Cov\left(\hat{\beta}_1,\hat{\beta}_2\right) = \left(\hat{\beta}_1 S_{xx}^2\right)^{-1}\left\{\sum_{i=1}^{n}\left(x_i-\bar{x}\right)\left(y_i-\bar{y}\right)\left[y_i-\bar{y}-\hat{\beta}_1\left(x_i-\bar{x}\right)\right]\left[y_i-\bar{y}-\hat{\beta}_2\left(x_i-\bar{x}\right)\right]\right\} \tag{14}$$

As a result, OLS Bisector technique gives a better performance since all the errors in the dependent variable and independent variable are included in the model (Saraçlı, 2008).

## 3. Application

The data set is obtained simulatively from MATLAB software considering different sample sizes (n=100, 250 and 500) and different distribution types (normal, Weibull and exponential). The performances of these two techniques are tested according to MAPE criteria which is given in (15).

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{e_t}{y_t}\right| \tag{15}$$

In ANN, data processing methods were used to increase the efficiency and performance of the training of the network. For this, the "Min-Max Normalization" method was used and it was calculated with the formula (16).

$$x' = (x_{max} - x_{min}) * \frac{(x_i - x_{min})}{(x_{max} - x_{min}} + x_{min} \quad (16)$$

Levenberg-Marquardt algorithm is used as training method to obtain the parameter estimates in ANN approach and for both techniques, 75% of simulated data is used for training while the 25% of data is used to test.

For the network architecture, trial and error was used. Between 1 and 100 trials were conducted for the second hidden layer. As a result of these experiments, 1 neurons in the first hidden layer and 10 neurons in the second hidden layer were used. Since there is no specific value in the literature about the learning rate, trial and error was used and the values of 0.01, 0.05, 0.001, 0.005 were applied. Since the value of 0.005 gives the best result, the learning rate was determined as 0.005.

While creating the ANN Bisector model, a model was established by determining the dependent variable as the input variable and the independent variable as the output variable. Then, the other model was established by determining the independent variable as the input variable and the dependent variable as the output variable. That is, the equations ANN(Y|X) and ANN(X|Y) have been calculated. While these equations were calculated through the MATLAB program, the curvilinear structure of the ANN was transformed into a linear structure by applying a transformation with the observation values. In this way, the regression equations of this linear structure were obtained.

According to different distributions and different sample sizes, analysis results are given in the Tables 1-3:

| **Method** | **ANN Bisector** | **OLS Bisector** |
|---|---|---|
| Errors | Performance (MAPE) | Performance (MAPE) |
| Normal | 0.1198 | 0.1256 |
| Weibull | 0.3058 | 0.3429 |
| Exponential | 3.3848 | 4.9237 |

**Table 1.** MAPE values for Type II bisector technique and ANN, n = 100.

According to Table 1, for n = 100, it is seen that the normal, Weibull, exponential distributions are tested in the ANN Bisector method and the MAPE values are 0.1198, 0.3085 and 3.3848, respectively. When the normal, Weibull and exponential distributions are tested in the OLS Bisector method, it is seen that the MAPE values are 0.1256, 0.3429 and 4.9237, respectively. Therefore, it is seen that ANN Bisector method gives better results than OLS Bisector method regardless of the distribution.

| **Method** | **ANN Bisector** | **OLS Bisector** |
|---|---|---|
| Errors | Performance (MAPE) | Performance (MAPE) |
| Normal | 0.1374 | 0.1421 |
| Weibull | 0.3121 | 0.3169 |
| Exponential | 18.6180 | 36.2912 |

**Table 2.** MAPE values for Type II bisector technique and ANN, n = 250.

When Table 2 is examined, while the data is distributed normally (1.0.2) and n = 250, the error of the artificial neural network model is MAPE = 0.1374 and the error of Type II bisector model is MAPE = 0.1641. While the data are distributed with Weibull 3.4 parameter, it is observed that the artificial neural network model is error MAPE =

0.3175 Type II bisector model is MAPE = 0.3353. While the data is distributed with 1 parameter exponential, artificial neural network model error is MAPE = 18.6180 and Type II bisector model error is MAPE = 12.8665. When Table 2 is summarized, it is clear that in the normal distribution and Weibull distributions, the artificial neural network model is more successful than the Type II bisector model. However, when the distribution of data is exponential, it is observed that the Type II bisector model is more successful than the artificial neural network model.

| **Method** | **ANN Bisector** | **OLS Bisector** |
|---|---|---|
| Errors | Performance (MAPE) | Performance (MAPE) |
| Normal | 0.1709 | 0.1738 |
| Weibull | 0.4142 | 0.4262 |
| Exponential | 3.2050 | 3.1202 |

**Table 3.** MAPE values for Type II bisector technique and ANN, n = 500.

When Table 3 is examined, while the data are distributed normally (1.0.2) and n=500, the error of the artificial neural network model is MAPE = 0.1709 and the error of Type II bisector model is MAPE = 0.1640. While the data were distributed with Weibull 3.4 parameter, it was observed that the artificial neural network model error was MAPE = 0.4142 Type II bisector model was MAPE = 0.3680. While the data are exponentially distributed with 1 parameter, the error of the artificial neural network model is MAPE = 13.2816 and the error of Type II bisector model is MAPE = 7.8151. When Table 3 is summarized, it is clear that in the normal distribution and Weibull distribution, the Type II bisector model is more successful than the artificial neural network model. However, when the distribution of data is exponential, it is observed that the artificial neural network model is more successful than the Type II bisector model.

## Conclusion

In case of any problem, if an estimate is to be made, the first thing that comes to mind is classical regression modeling. However, it is not possible to provide the assumptions of classical regression techniques in real life. In cases where such assumptions are not fully satisfied, it would be the right choice to consult flexible calculation methods. Considering its advantages in this regard, it may be the right choice to use ANN as it does not have compelling assumptions like classical techniques. However, since there may be measurement errors in real-life data and considering the advantage of adding the error in the independent variable to the model, it seems that using Type II regression techniques can be a strong alternative for researchers.

In this study, when we examine the earlier studies in the literature, there are some studies which used ANN approach to estimate the regression, time series and etc. parameters. There are also some studies which calculates the regression line considering the error term for the independent variables. The importance of this study is: here we combined ANN and Type II regression techniques and calculate the ANN bisector regression line. The results showed that in many situations (like for different types of distributions and sample sizes) the performance of ANN-Bisector regression is better than the OLS-Bisector regression according to MAPE values. One of the other importance of this study is to mention about the superioty of ANN in Type II regression analysis and for the future studies apply ANN approach in Type II regression to obtain better results than earlier studies.

In this research, it is seen that as a result of combining classical techniques and flexible calculation methods, the models desired to be obtained work more robustly and with low error. It is predicted that the obtained ANN Bisector technique can be used as a very strong alternative, both because it includes the errors of the OLS Bisector technique due to the independent variable, and because of the advantages of not having difficulty with ANN under assumptions.

## References


Aygören, H., Sarıtaş, H., Moralı, T. (2012). İMKB 100 Endeksinin Yapay Sinir Ağları ve Newton Nümerik Arama Modelleri ile Tahmini. Uluslararası Alanya İşletme Fakültesi Dergisi, 73-88.

Bayır, F. (2006). Yapay sinir ağları ve tahmin modellemesi üzerine bir uygulama. Yüksek Lisans Tezi, İstanbul Üniversitesi Sosyal Bilimler Enstitüsü. İstanbul.

Dawson, C. W., Wilby, R. (1998). An artificial neural network approach to rainfall-runoff modelling. Hydrological Sciences Journal. 43(1), 47-66.

Ergezer, H., Dikmen, M., Özdemir, E. (2003). Yapay sinir ağları ve tanıma sistemleri. PIVOLKA, 2(6), 14-17.

Gazeloğlu, C., Saraçlı, S. (2013). A Simulation Study on Nonlinear Type II Regression Analysis. IAAOJ, Scientific Science, 1(1): 13-18.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. Proc. Nat. Acad. Sci. 79, 2554-2558.

Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. Proc. Nat. Acad. Sci. 81, 3088-3092.

Isobe, T., Feigelson, E.D., Akritas, M.G. and Babu, G.J. (1990). Linear Regression in Astronomy I, the Astrophysical Journal, 364, 104-113.

Khan, K., Sahai, A. (2012). A Comparison of BA, GA, PSO, BP and LM for Training Feed Forward Neural Networks in e-Learning Context. I.J. Intelligent Systems and Applications, 7, 23-29.

Matlab User's Guide, (2020). https://www.mathworks.com/help/deeplearning/ ref/logsig.html#:~:text=logsig%20is%20a%20transfer%20function,output%20from%20its%20net%20input.&text=and%20returns%20A%20%2C%20the%20S,A%20with%20respect%20to%20N%20. Date of Access: 28.10.2020.

Rumelhart, D. E., Hinton, G. E., Williams, R. J. (1986). Learning Internal Representation by Error Propagation. In: Rumelhart DE, McClelland JL and the PDP Research Group (eds) Parallel distributed processing, 1. MIT Press, Cambridge, Mass.

Saraçlı S. (2008). Comparison of Linear Regression Techniques in Measurement Error Models -Monte-Carlo Simulation Study -. Doctoral Dissertation, Eskisehir Osmangazi University, Department of Statistics.

Saraçlı, S. (2011). Tip II Regresyon Tekniklerinin Monte-Carlo Simülasyonu ile Karşılaştırılması. e-Journal of New World Sciences Academy, 6(2): 26-35.

Saraçlı, S., Türkan, A.H. (2013). A Comparison of Linear Regression Techniques In Method Comparison Studies. Journal of Statistical Computation and Simulation, 83(10): 1890-1899.

Saraçlı, S., Yılmaz, V. and Doğan, I. (2009). Simple Linear Regression Techniques in Measurement Error Models (Review). Anadolu University Journal of Science and Technology, 10: 335-342.

Saylor, R.D., Edgerton, E.S. and Hartsell, B.E. (2006). Linear regression techniques for use in the EC tracer method of secondary organic aerosol estimation., Atmospheric Environment 40, 7546 -7556.

Suresh, S., Omkar, S. N., Mani, V. (2005). Parallel Implementation of Back-Propagation Algorithm in Networks of Workstations, IEEE Transactions on Parallel and Distributed Systems, 16(1), 24-34.

Tunca, B. (2019). Artificial neural network approach on type II regression analysis. M.Sc. Thesis, Afyon Kocatepe University Institute of Science, Afyonkarahisar.

Uğur, A., Kınacı, A. C. (2006). Yapay Zeka Teknikleri ve Yapay Sinir Ağları Kullanılarak Web Sayfalarının Sınıflandırılması. "Türkiye'de İnternet" Konferans Bildirileri. Ankara: TOBB Ekonomi ve Teknoloji Üniversitesi. 345-349.