Forecasting Students' Enrollment Using Neural Networks and Ordinary Least Squares Regression Models

Egbo, M.N[1] and Bartholomew, D.C[2]

[1]Department of Statistics, Federal University of Technology, Owerri, Imo state, Nigeria,
[2]Leadway Assurance Company ltd, Iponri, Lagos State, Nigeria
Email: egbomary4u@yahoo.com, bartholomewdesmond@gmail.com

Abstract: Based on the presentation of dynamic and nonlinear data forecast, we discuss the difference in two approaches, Multi-layer feed-forward artificial neural networks and ordinary least squares regression for students' enrolment forecast in FUTO, Nigeria. A simple procedure to include the Mean square error (MSE), Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) is proposed and tested. The result suggested that the Multi-Layer Feed-Forward Artificial Neural Networks provides better predictions for nonlinear and chaotic systems.

Keywords: Forecasting, mean square, error, neural network, least squares.

1 Introduction

According to [1], enrollment projection provides information for decision making and budget planning. Consequently, students' population and enrollment number should be considered as a chaotic system. In this case, some small change in one of the conditioning factors may bring in a sizeable change in the enrollment figure of students at a given point in time. Methods such as statistical procedures, conventional mathematical and artificial intelligence (AI) techniques have been used in enrollment projection. [2] applied the fuzzy time-series model and neural network approaches to predict university students' enrollments and the results found to be satisfactory than the equivalent statistical and mathematical methods. [3] Stated that statistical approaches are not very suitable for predicting chaotic system like students' enrollment because they make assumptions, which are sometimes found unrealistic and cannot deal with the inbuilt chaos.

According to [4] Artificial Neural Network can be defined as a mathematical model for predicting system performance (output) inspired by the structure and function of biological neural network. In the 1940s, McCulloch and Pitts found that a neuron can be modeled as simple threshold device to perform logic function.

The Hopfield model introduced in 1982 ushered in the modern era of neural network research. The model works at the system level rather than at a single neuron level. This network can be used as an associative memory for information storage and for solving optimization problems.

In the last decade, Artificial Neural Network has been widely used for forecasting data of dynamic and non-linear data in nature. Many previous studies have been discussed for forecasting using artificial neural network such as enrolment, the stock index, temperature and financial forecasting, etc.

[5] applied Neural Network and Auto-regressive Integrated Moving Average (ARIMA) in investigating the Neural Network for the linear time series forecasting. He found out that Neural Network outperforms than linear time series model. [6] used Artificial Neural Network (ANN) in forecasting the river flow in the U.S.A. They compared the result of the model with that of Auto-regression (A.R). It was found that ANN model perform better than that of auto-regression model. While further studies in [7] suggested that ANN is best suitable for forecasting water demand in irrigation districts. [8] predicted the population of India with ANN model using demographic data collected from "International Brief: Population trends in India".

Also [9] applied artificial neural network and multiple regression analysis in forecasting stream water temperature. [10] predicted the river flow base on developed discrepancy ratio with Artificial Neural Network and principle component analysis multivariate linear regression model. The result shows that ANN is more accurate than regression but more complex while regression is quite simple.

Copyright © 2018 Isaac Scientific Publishing

Recent works from year 2010 also showed that system for predicting the population of Nigeria is best estimated using ANN model [11]. [12] also applied Artificial Neural network and multiple regression techniques for container at Bangkok ports of Thailand. The results showed that ANN has a mean square error of 4814.7768 while that of multiple regressions was 6028.5571. This also proposes that ANN model has a better predicting performance than multiple regressions. [13] compared the forecasting capabilities of Artificial Neural Network and traditional ARIMA model using the A-1 Quds index of stock exchange of Palestine. The results showed that the ANN model provides a satisfactory result than the ARIMA model. [14] used artificial neural network in predicting students' enrollment of Tia Solarin University of Education Ogun state Nigeria. They compared the mean absolute error of the model which was 0.0101 percent with that of linear regression model and auto regression which were 0.0570 percent and 0.0725 percent respectively.

This paper covers prediction of students' enrollment in the Federal University of Technology, Owerri (FUTO), Imo state, Nigeria. The objective is to display a model of multi-layer feed-forward artificial neural network that will forecast students' enrollment in FUTO and compare the model prediction accuracy.

2 Multilayer Feed-Forward Neural Networks (MLP)

Networks that contain more than one layer of artificial neurons, which allows only unidirectional forward connections of inputs and outputs, are called multi-Layer Perceptions (MLP) or multi-layered feed-forward Neural Networks. The underlying structure of a MLP consists of neurons and synapses [16]. The neurons are organized in layers, which are usually fully connected by synapses. The input layer consists of all covariates in separate neurons and the output layer consists of the response variables. The layers in between are referred to as hidden layers. Input layer and hidden layers include a constant neuron relating to intercept synapses or bias, (i.e. synapses that are not directly influenced by any covariate called the intercept). Figure 1 below gives an example of a neural network with one hidden layer that consists of three hidden neurons. This neural network models the relationship between the two covariates A and B and the response variable Y and it is called a MLP feed-forward Neural Network.

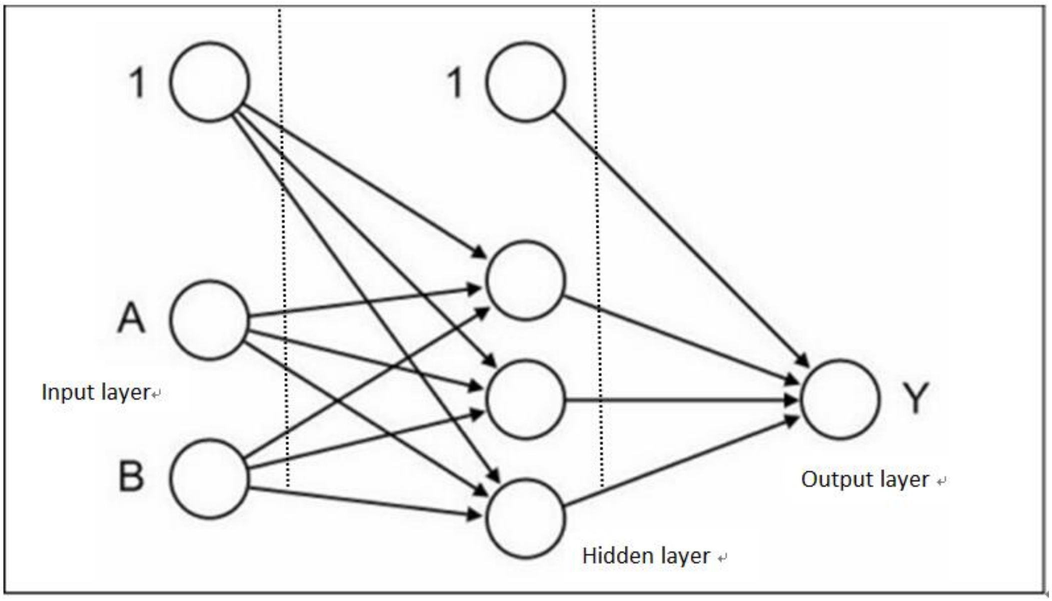

Figure 1. Multi-layer feed-forward neural network.

2.1 Activation Function

The activation function $\varphi\left(v\right)$ also called the transfer function is the mechanism by which the artificial neuron processes information and passes it throughout the network. It determines the relationship between inputs and outputs of a node and network in neural network model. In practice, only threshold function, sigmoid (logistic) function and the piecewise linear function are used.

2.2 Estimation of Parameters

Poor estimate of the parameters occurs when the function is wrongly selected [13]. Therefore, the builders of Artificial Neural Networks are interested in the shape function, as specified by the magnitude of the functional parameters. Parameters of the Artificial Neural Networks can be estimated by using the methods of learning algorithms. The result of these learning algorithms depends on whether or not the transfer function is differentiable. The popular common nonlinear functions used are the tangent hyperbolic function and the sigmoid function. For this research, we used only the sigmoid (logistic function) which is preferable when a network contains hidden layers. For non-differentiable activation functions, the error obtained during the training process (which serves in the estimation of parameters) is constant, hence the gradient does not exist or equals zero, and thus the learning process is impossible when some learning algorithms, such as back-propagation are used [13]. By using the sigmoid (logistic) function it is possible to tell whether the change in weights is good or not, because a small change in the weights will generate some change in output [13]. With the step unit function, a small change in weights will usually generate no change in output [16].

2.2.1 The back-propagation learning algorithm

The algorithm that uses the gradient descent learning technique for multilayer feed-forward neural networks is known as Back-Propagation, or generalized delta rule as stated by Rumelhart (1986). Training network consists of two main parts, the input and output parts. Initially all weights are usually initialized with random values which may be drawn from a standard normal distribution or uniform distribution [16].The network outputs depend on the input units, hidden units, weights of the network, and the activation function. The difference between the predicted and the observed output (target) is known as network error. Back-propagation method takes the network error and propagates it backward into the network. Errors are used at each neuron to update weights. This process is repeated until the total network error becomes the smallest.

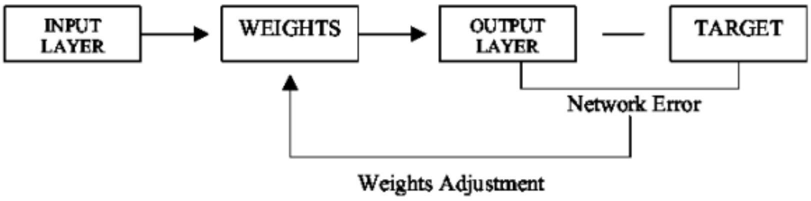

Figure 2. Feed-forward back-propagation technique

Neural networks method uses the mean squared error, root mean squared error or cost function to measure the difference between the target value and the output value. Weights of the network are frequently adjusted in such a way that the error or objective function becomes as small as possible.

2.3 The General Artificial Neural Network Model

$$v_k = \sum_{j=1}^{m} w_{kj} x_j \tag{1}$$

$$y_k = \varphi\left(v_k + b_k\right) \tag{2}$$

where:

$x_1, x_{2,} \dots, x_m$ is a set of inputs (explanatory variables).

$w_{k1}, w_{k2,} \dots, w_{km}$ is the set of synaptic weights excluding the bias (coefficient of the model parameters)

v_k is linear combination of output due to the input signals.

b_k is bias which has weight w_{k0} (intercept term)

$\varphi(.)$ is activation function.

y_k is output (response variable).

The bias is modeled as an input to the neuron, the effect of the bias is accounted by adding a covariate signal fixed at+1and adding a synaptic weight equal to the bias b_k.

2.3.1 Multi-layered feed-forward neural network

$$Y_k(x) = \varphi\left(w_o + \sum_{j=1}^{J} w_j.\varphi\left(w_{oj} + \sum_{j=1}^{m} w_{ij}.x_i\right)\right)$$
$$= \varphi\left(w_o + \sum_{j=1}^{J} w_j.\varphi\left(w_{oj} + w_j^T x\right)\right) \tag{3}$$

where:

w_o is intercept of the output neuron.

w_{oj} is intercept of the jth hidden neuron.

w_j is synaptic weights corresponding to the weights starting at the jth hidden neuron and leading to the output neuron.

w_{ij} is vector of all synaptic weights corresponding to the i^{th} synapses leading to the j^{th} hidden neuron,

$x = (x_1, ..., x_m)$ is vector of all explanatory variables.

Although more hidden units in the system normally result in better forecast, more hidden units may lead to over-fitting on sample data. Limit number of hidden layers is based on trial and error.In this paper, we will make use of one hidden layer.

2.3.1.a Activation function

For the purpose of this study, sigmoid activation function is the will be used. Because of its differentiable function made importance in the training of neural networks. The logistic function whose domain is [0, 1], can be represented as:

$$\varphi(v) = \frac{1}{1+\exp(-av)} \quad \begin{cases} a > 0 \\ -\infty < v < \infty \end{cases} \tag{4}$$

where a is the slope parameter of the sigmoid function, and v is the sum of all the inputs and the bias term. By varying the parameter a, we obtain sigmoid functions of different slopes, as illustrated below.

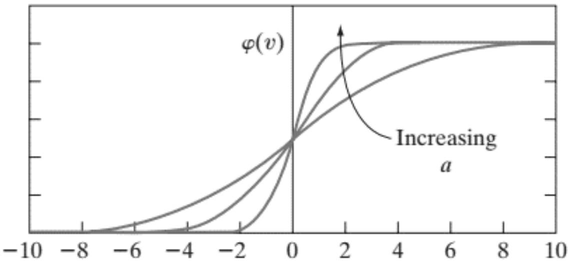

Figure 3. Sigmoid function for varying slope parameter a.

The logistic function turns on gradually when activated. When the logistic function approaches 0, the function is almost insensitive to impulse received from input layer, meaning output is inactivated (Yassen, 2011).

2.3.1.b Evaluation of the accuracy of measurement

Neural networks method uses the mean squared error, root mean squared error or cost function to measure the difference between the target value and the output value. Weights of the network are frequently adjusted in such a way that the error or objective function becomes as small as possible. The target function can be written as:

$$E_k = T_k - Y_k \tag{5}$$

where T_k is the actual output value of the t^{th} iteration (observed output), and Y_k is the computed output of the t^{th} iteration (predicted output). The most common cost functions used are the root-mean squared error (RMSE) and the mean squared error (MSE). The root- mean squared error function can be stated as:

$$RMSE = \sqrt{MSE = \frac{1}{N}\sum_{t=1}^{N}\left(T_k - Y_k\right)^2} \tag{6}$$

However the mean squared error (MSE), is expressed as:

$$MSE = \frac{1}{N}\sum_{K=1}^{N}(T_K - Y_K)^2 \tag{7}$$

$$MAPE = \frac{100}{n} \mid \sum_{i=1}^{n}(\frac{T_k - Y_k}{T_k}) \mid \tag{8}$$

The network is most commonly trained using the MSE error function. But in ANN modeling this differs slightly from Eq.7 where the network objective function is:

$$E = \frac{1}{2}\sum_{t=1}^{N}(\mathrm{T}_k - \mathrm{Y}_k)^2 \tag{9}$$

The constant ½ is used to facilitate the computation of the derivative for the error function, which is essential in the estimation of the parameters. In the derivative of Eq.9 the constant ½ disappears, while if eq.7 is used, 1/N could not be reduced and we would end up with the factor 2/N; hence making it hard to determine the network parameters.

We can say that back-propagation approach consists of four steps:

i. The Algorithm determines a random value of initial parameters θ_0

ii. The calculation of the difference between the output and the target value. *The gradient of the error function is propagated backward respectively from the output layer to hidden layer.

iii. Updating the network weights to reduce the error using Equation (3.24).

iv. Repeat the second and the third steps until the gradient of the error function gets close to zero, if $\theta_{k+1} \approx \theta_k$, since the optimality condition is that $\nabla E\left(\theta\right) = 0$.

The above process of four steps is known as the epoch or threshold (parameter to stop the training process). Thus, the following threshold was considered in the study; 0.1, 0.05 and 0.01.

2.3.1.c Proposed ANN model for students'' enrollment

Here, number of students enrolled is the response or target that we wish to predict, we refer to it as $Number.\mathrm{E}nrolled$, number of JAMB applicants (predictor) as $Number.Jamb.\mathrm{Applicants}$, Post UTME performance and Graduate output as $Post.Utme.Performance$ and $Graduate.Output$ respectively.

We propose an artificial neural network model (ANN) for our research study as:

$$\hat{Y} = w_0 + w_1 Number.Jamb.Applicant_t + w_2 Putme.Performance + w_3 Graduate.\mathrm{Output}_{\mathrm{t}} + \varepsilon_t. \tag{10}$$

where:

$\hat{Y} = Number\ Enrolled$

$w_0 =$ Bias (intercept term)

w_1, w_2, w_3 is synaptic weights.

2.4 Multiple Linear Regressions

Linear regression analysis is a statistical technique for modeling the linear relationship between two or more variables. It used to explain the relationship between a dependent variable and two or more independent variables in such a way that the behavior of dependent variable can be predicted from the independent variables if the relationship exists.

The model is given as:

$$y = \beta_0 + \sum_{j=1}^{k} \beta_j X_j + \varepsilon_j \tag{11}$$

where
Y = response variable
$\beta_0, \beta_1, ..., \beta_k$ is partial regression coefficients
X is predictor variables
ε is error term

The parameters β_j represents the expected change in the response y per unit change in the X_j when all the remaining independent variables $X_i \left(i \neq j\right)$ are held constant. The term 'linear' because the eq.11 is a linear function of the unknown parameters $\beta_0, \beta_1, ..., \beta_K$.

2.4.1 Estimation of model parameters (matrix approach)

$$Y = X\beta + \varepsilon \tag{12}$$

where Y is an nx1 vector of response variable, X is an n*(k+1) matrix of explanatory variables, β is a (k+1)-dimensional column vector, and ε is an n-dimensional column vector .

Thus, we apply the method of ordinary least square to show theoretically how to obtain the estimator of β .

The Ordinary Least Square (OLS) seek an estimate of β that minimizes the sum of squared deviations between the fitted and the observed responses. Thus, we wish to find the value of β that minimizes

$$Q\left(\beta\right) = \sum_{i=1}^{n} \varepsilon_i^2 = \varepsilon'\varepsilon = \left(Y - X\beta\right)' \left(Y - X\beta\right) \tag{13}$$

The least square estimate, $\hat{\beta}$ of β is therefore the solution of β in the equation $\frac{\delta Q}{\delta\beta} = 0$.

$$\begin{aligned} Q\left(\beta\right) &= Y'Y - \beta'X'Y - Y'X\beta + \beta'X'X\beta \\ &= Y'Y - 2Y'X\beta + \beta'X'X\beta \end{aligned} \tag{14}$$

Since $\beta'X'Y$ is a 1x1 matrix or a scalar, and its transpose $\left(\beta'X'Y\right)' = Y'X\beta$ is the same scalar. The least squares estimate of β must satisfy:

$$\frac{\delta Q}{\delta_\beta}\Big|_{\hat{\beta}} = -2X'Y + 2X'X\hat{\beta} = 0 \tag{15}$$

which simplifies to:

$$X'X\hat{\beta} = X'Y \tag{16}$$

Eq.16 is the OLS normal equation in matrix form. To solve the equation, pre-multiply both side of eq.22 by the inverse of $X'X$. This gives the least squares estimate of β as:

$$\hat{\beta} = \left(X'X\right)^{-1} X'Y \tag{17}$$

The solution assumes that the matrix $X'X$ is nonsingular and so $\left(X'X\right)^{-1}$ exist. $\left(X'X\right)^{-1}$ exists if the regressors are linearly independent. That is if no column of X is a linear combination of the other columns.

Having obtained the estimators of β which is $\hat{\beta}$, the fitted multiple linear regression model is:

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2, ..., \hat{\beta}_k x_k + \varepsilon_i \tag{18}$$

And the difference between the observed y_i and the predicted $\hat{y}_i$ is the residual error given by:

$$\varepsilon_i = y_i - \hat{y}_i \tag{19}$$

2.4.2 Proposed MLR model for enrollment

Multiple Linear Regression Model is given by

Copyright © 2018 Isaac Scientific Publishing

$$\hat{Y} = \beta_0 + \beta_1 Number.Jamb.Applicants + \beta_2 Putme.Performance + \beta_3 Graduate.Output + \varepsilon_t . \tag{20}$$

$\hat{Y}$ is *Number Enrolled*

β_0 is intercept

$\beta_1, \beta_2, \beta_3$ is beta coefficient of the various independent variables.

ε_i is error

2.4.3 Normalization

Transformation using min-max normalization was applied on the original data. The function is given by:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{21}$$

where:

x' is the data after normalization.

x is the current data to normalize.

$\min(x)$ is the minimum value of the data to be normalized.

$\max(x)$ is the maximum value of data to be normalized.

We note that any transformation applied to the data prior to training the model will have to be applied in reverse later on. In order to convert back to the original units, R provides the rescaling function which is given by:

$$x = [x - \min(x')] * [\max(\mathrm{x'}) + \min(\mathrm{x'})] \tag{22}$$

It is also to note that the same scaled data was used for both ANN model and multiple regression to enable equal comparison of error and model performance.

3 Results and Discussion

3.1 Illustrative Example

The data was obtained from the academic and university admission unit of Federal University of technology, Owerri. The data are based on the number students' enrolled yearly, number of students that applied for FUTO in University Tertiary Matriculation Examination (UTME), Post UTME performance and Graduate output from 2006 to 2016.

Table 1. Enrollment strength from 2007/2008 to2016/2017 Academic session

Academic Session	Number of students Enrolled	Number of JAMB Applicants	Number of students that wrote PUTME	Number of students that scored 180+	PUTME pass performance	Graduate Output
2007/2008	2,041	14,283	11,532	6,573	57	1,861
2008/2009	3,229	19,685	15,449	9,958	64	2,432
2009/2010	4,271	19,880	15,267	11,045	65	4,106
2010/2011	5,352	32,453	13,324	9,344	65	5,467
2011/2012	3,352	28,467	34,252	21,135	64	5,234
2012/2013	4,973	45,549	38,456	26,896	70	3,034
2013/2014	5,074	49,231	11,276	8,220	71	2,130
2014/2015	4,973	34,532	12,276	8,588	70	2,127
2015/2016	5,314	34,876	15,763	12,922	82	2,225
2016/2017	5,445	39,013	17,927	15,670	87	2,243

Source: Academic planning and development, FUTO.

Multiple Regression Model and Model Diagnostic.

$H_0 : \beta_1 = \beta_2 = \ldots\ldots\ldots\ldots\ldots = \beta_i = 0$

$H_1 : \beta_i \neq 0$ at least for one i

	estimate	*std error*	*t-value*	*p-value*		
Intercept	-256.8	200.3	-128.2	0.92472		
Jamb. Applicant	0.0530	0.0223	2.539	0.056		
PUTME performance	0.6710	0.3002	2.234	0.0669		
Graduate output	0.1930	0.1413	1.366	0.2211		
Residual Standard error	Degree of freedom		Multiple R	Adj R^2	F-stat	p-value
0.1845014	6.00		0.80	0.78	8.128498	0.023

Since *p-value*: 0.023 is less than 0.05 alpha value, we reject the H_o and conclude that the intercept is not equal to zero and the regression model is significant.

R-squared of 0.80 explains that 80% variation of number of students being enrolled could be explained by the model.

The proposed model:

$$\hat{Y} = -256.8 + 0.053NO.JAMB.APP_t + 0.671PUTME.PEF_t + 0.193GRAD.OUT_t + \varepsilon_t.$$

Modeling with ANN model.

The neural net package by Stefan Fritsch and Frauke Guenther provides a standard and easy-to-use implementation of such networks. It also offers a function to plot the network topology. For these reasons, the neural net implementation is a strong choice for learning more about neural networks.

The multilayer feed-forward network was trained with only a single hidden node.

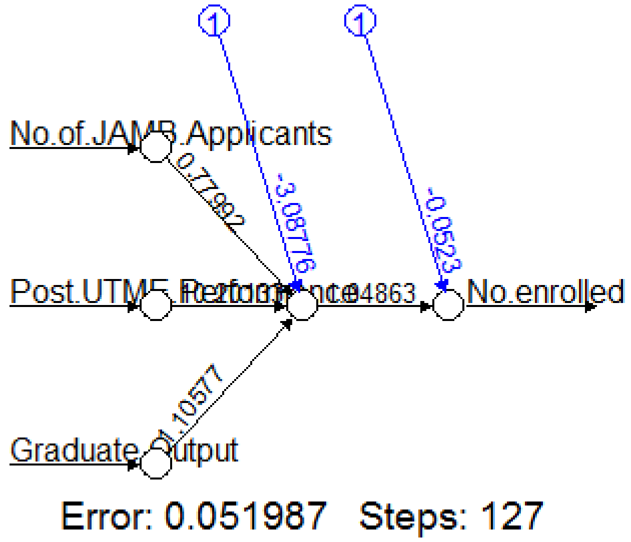

Figure 4. The result topography of the proposed ANN model.

The model error is 0.051987 and it was iterated 127 times.

The proposed ANN model is given as:

$$\hat{Y} = -3.0878 + 0.7791NO.JAMB.APP_t + 1.0203PUTME.PEF_t + 1.1058GRAD.OUT_t + \varepsilon_t.$$

Forecasting with regression model

Table 2. The predicted values of the model using regression model.

Academic section	Actual enrollment value (T_k)	Predicted enrollment value (y_k)	Error| $E_i = (T_k - Y_k)$
2007/2008	2041	2447	-406
2008/2009	3229	3872	-643
2009/2010	4271	5121	-850
2010/2011	5352	6417	-1065
2011/2012	3352	4025	-673

Copyright © 2018 Isaac Scientific Publishing

2012/2013	4973	5962	-989
2013/2014	5074	6083	-1009
2014/2015	4973	5962	-989
2015/2016	5314	6372	-1058
2016/2017	5445	6528	-1083

Forecasting with Artificial Neural Network

Table 3. The predicted values of the model using ANN

Academic section	Actual enrollment value y_i	Predicted enrollment value $\hat{y}_i$	Error ε_i
2007/2008	2041	2019	22
2008/2009	3229	3194	35
2009/2010	4271	4225	46
2010/2011	5352	5295	30
2011/2012	3352	3321	31
2012/2013	4973	4920	53
2013/2014	5074	5020	54
2014/2015	4973	4920	53
2015/2016	5314	5257	57
2016/2017	5445	5387	58

Evaluation of the accuracy of the models

Mean square error

$$Mean\ square\ error = \frac{1}{10}\sum_{1}^{10}\left(T_k - Y_k\right)^2$$

For multiple regression

$$\left(406^2+643^2+850^2+1065^2+673^2+1009^2+989^2+1058^2+1083^2\right)*\frac{1}{10}=\mathbf{8155.5}$$

For Artificial Neural Network

$$MSE = {}^{1}\!\!\diagup_{10}\left(22^2+35^2+46^2+30^2+31^2+53^2+54^2+53^2+57^2+58^2\right)*\frac{1}{10}=2083.3$$

Root mean square error

$$RMSE=\sqrt{\sum_{i=1}^{10}(\mathrm{T}_k-Y_k)^2} = \sqrt{MSE}$$

For multiple regressions:

Root mean sum of error= $\sqrt{8155.5}$ =903.3.

For ANN: $RMSE = \sqrt{2083}$ =45.6

Mean absolute percentage error (Mape)

$$MAPE=\frac{100}{10}\,|\sum_{i=1}^{10}(\frac{T_k-Y_k}{T_k})\,|$$

For multiple regression:

Table 4. Absolute percentage error of multiple regression model

Academic section	Actual enrollment value (T_k)	Predicted enrollment value (y_k)	Error $E_i=(T_k-Y_k)$	$\lvert\frac{T_k-Y_k}{T_k}\rvert$
2007/2008	2041	2447	-406	0.1989
2008/2009	3229	3872	-643	0.1991
2009/2010	4271	5121	-850	0.1990

Copyright © 2018 Isaac Scientific Publishing

2010/2011	5352	6417	-1065	0.1990
2011/2012	3352	4025	-673	0.2008
2012/2013	4973	5962	-989	0.1989
2013/2014	5074	6083	-1009	0.1988
2014/2015	4973	5962	-989	0.1989
2015/2016	5314	6372	-1058	0.1991
2016/2017	5445	6528	-1083	0.1989

$$MAPE = \frac{100}{10} * 1.99142 = 19.9142$$

Table 5. Absolute percentage error of ANN MODEL

Academic section	Actual enrollment value (y_i)	Predicted enrollment value ($\hat{y}_i$)	Error (Ei)	$\mid \frac{T_k - Y_k}{T_k} \mid$
2007/2008	2041	2019	22	0.0101
2008/2009	3229	3194	35	0.0101
2009/2010	4271	4225	46	0.0108
2010/2011	5352	5295	30	0.0107
2011/2012	3352	3321	31	0.0056
2012/2013	4973	4920	53	0.0092
2013/2014	5074	5020	54	0.0106
2014/2015	4973	4920	53	0.0107
2015/2016	5314	5257	57	0.0107
2016/2017	5445	5387	58	0.0106

$$MAPE = \frac{100}{10} * 0.10056 = 1.0056$$

Table 6. The values of the error types for the models.

Error	ANN	MLR
MSE	2083.3	8155.5
RMSE	45.6	903.3
MAPE	1.006	19.914

4 Conclusion

Artificial Neural Network is a better forecasting model than multiple regression models because it gives a minimum error value. The benefit of the model is that, it will help the school management to plan adequately for the future needs of the learners in provision of facilities. Also, government, public and private sectors will make reliable and enduring plans that will benefit both the planners and the beneficiaries.

5 Recomendation

We recommend that the management of various higher institution, government, public and private sectors to take advantages of the high-capacity of artificial neural networks as a forecasting technique in various fields, such as medical research, genetics research, industrial research, energy, and military research etc. We also recommend that Statisticians should adopt ANN approach as a method of Statistical prediction and forecasting as well as conducting further research on neural networks to investigate their suitability for use in other statistical applications.

Copyright © 2018 Isaac Scientific Publishing

References

1. Guo, S.,(2002)."Three Enrollment Forecasting Models: Issues in Enrollment Projection for Community Colleges". Presented at the 40thRP Conference, May 1-3 2002, Asilomar Conference Grounds, Pacific Grove, California.
2. Song, Q.,& Chisom, B.S.,(1993)."New Models for Forecasting Enrollments: Fuzzy Time Series and Neural Network Approaches".A Paper presented at the Annual Meeting of the Mid-south Educational Research Association,20th Lexinton, K Y,12-15.
3. Keilman, N., Pham, D.Q., Hetland, A.,(2002). "Why Population Forecasts should be probabilistic" Illustrated by the case of Norway. Demographic Research 6(15): 409-453.
4. Andrej, K.,Janez, B. and Andrej,K. (2011). "Introduction to the Artificial Neural Networks". Intech Publishing, Unit 405, Office Block,Hotel Equatorial Shanghia China.
5. Zhang, G.P.(2001)."An investigation of neural network for linear time series forecasting". Journal of Computer and Operational Research. Vol.22, pp 1183-1202.
6. Baareh, A.M.,Sheta, A.F. and Khnaifes, K.A. (2006). "Forecasting River flow in the United State of America. A comparison between Auto-regression andNeural Network Non-parametric models" Journals of Computer science. Vol. 2, pp 775-780.
7. Palido-Calvo,I., Montessions,P., Roldan, J. and Ruiz-Navarro, F. "Linear regression and neural approaches to water demand forecasting in irrigation districts with telemeting systems". Journal of Biosystems Engineering. Vol. 97, pp.283-293.
8. Bandyopadhyay, B.,& Chattopadhyay, S.(2008). "An Artificial Neural Net approach toForecast the Population of India".1/9 Dover Place,Kolkata700019,WestBengal,Idian.E-maIL:surajit_2008@yahoo.co.in
9. Sahoo, G. B., Schladow, S. G. and Renter, J. E. (2009). "Forecasting Stream Water Temperature using Regression Analysis, Artificial Neural Network and Chaotic non-linear dynamic models". Journal of Hydrology. Vol. 378 pp325-342.
10. Noori, R. (2010). Comparison of Artificial Neural Network and Principle Component Analysis Multivariate Linear Regression models for predicting the river flow based on developed discrepancy ratio statistics. Journal of Expert System. Vol. 37, pp. 5856-5862.
11. Folorunso, O., Akinwale, A.T., Asiribo,O.E., & Adeyemo, T.A.(2010). Population Prediction Using Artificial Neural Network. Africa "Journal of Mathematics and Computer Science Research" Vol.3(8). Pp.155-162.
12. Gosasang, V., Chanraprakaikul,W. and Kiattsi,S.(2011)."A Comparison of Traditional and Neural Networks Forecasting Techniques for Container Throughput at Bangkok Port". The Asian Journal of Shipping and Logistics. Vol. 27 Number 3 December 2011 pp. 463-482.
13. Yassen, A.A. (2011). "Comparative Study of Artificial Neural Network and ARIMA Models for Economic Forecasting". A thesis submitted in partial fulfillment of requirements for the degree of Master of Statistics, Al-Azhar University Gaza.
14. Rufia,F.A., Folorunso,A. & Usma,O.L. (2015)."Predicting Students' Enrollment Using Generalized Feed-Forward Neural Network".Journal of Natural Science, Engineering and Technology 2015. Vol. 14(1):pp. 58-70.
15. Cheng.B and Titterington, D.M (1994)." Neural Network: A Review from a Statistical Perspective". Journal of Statistical Science,
16. Haykin, S.(1999). Neural Networks: a comprehensive foundation. 2nd edition, Pretence Hall, Upper Saddle River, N.J.
17. Brooks. (2010). Introduction to statistical economics. Cambridge press.
18. Academic planning and development unit (2017). Students Statistical Reports and Information, Federal university of technology, Owerri, Imo state, Nigeria.
19. Ben, K. and Patrick V. S. (1996). "An Introduction to Neural Network". University of Amsterdam Publication, Kruislaan 403, NL1098 SJ Amsterdam Netherlands.
20. Brett, L. (2015). "Machine Learning With R. Second Edition". Packet Publishing, Birmingham Mumbai. Institute of Mathematical Sciences, University of Kent England. Vol9, pp1-54.
21. Du, K. L. and Swamy,M.N.S.(2013). "Neural networks and statisticallearning". https://www.researchgate.net/publication/264912450
22. Frauke, G. and Stefan, F.(2015). "Neural net: Training of Neural Networks". The R Journals. Vol2/1 pp30-38.

Copyright © 2018 Isaac Scientific Publishing

23. Guoqiang, Z., Patuwo, B .E .and Hu, M. Y. (1998)."Forecasting with artificial neural networks: The state of the art". International Journal of ForecastingVol14 (1998) pp35–62.

24.Https//en.on.wikipipedia.org/wiki/artificial_neural_network/t.echop.

25. Kaastra, I. and Boyd, M. (1996). "Designing a Neural Network for Forecasting Financial and Economic Time Series". Journal of Neuro-computing Vol. 10 pp.215-236.

26. Mehta, A.C. (2004). "Projections of Population, Enrolment and Teachers". Module on enrollment and populations. National Instituteof Educational Planning and Administration, 17-B, Sri Aurobindo Marg, New Delhi-110016 (INDIA), January 18, 2004.

27. Onimode, B.M., Alhassan, J.K. & Adepoju, S.A. (2015). "Comparative Study of Inflation Rates Forecasting Using Feed-Forward Artificial Neural Networks and Auto Regressive (AR) Models".International Journal of Computer Science Issues. Volume 12, pg 260-266 Issue 2, March 2015.

28. Robert, I.K. (2011) .R in Action: Data Analysis and Graphics with R. Manning Publications Co. 20 Baldwin Road PO Box 261 Shelter Island, NY 11964.

29. University Admission Unit. (2017). Students Statistics. Federal University of Technology, Owerri, Imo state, Nigeria.

APPENDIX

```
#Import the data into R
No.stdnt.Enrolled<-read.csv("C:/Users/pc/Desktop/Robby project/analysisdata.csv")
#edit the variable names
fix(No.stdnt.Enrolled)
 The structure of the data was checked.
str(No.stdnt.Enrolled)
data. Frame':
$ Enrolled.Stdnt
$ Jamb.Applicant
$ PUTME.Performance
$ Graduate.Output :num 1861 2848 4193 5948 5950 ..
3. summary(No.stdnt.Enrolled$Enrolled.Stdnt)
normalize<- function(x) {return((x - min(x)) / (max(x) - min(x)))}
shapiro.test(lm.fit$)
data: lm.fit$residuals
durbinWatsonTest(lm.fit)#HOMOSCEDASTICITY
ncvTest(lm.fit)
Variance formula: ~ fitted.values
> #As a general rule, if sqrtvif
#Store the fitted result output of the ANN model
Out<cbind(Student.enrollment_model$covariate,
Student.enrollment_model$net.result[[1]])dimnames(out) <-
list(NULL,c("No.of.JAMB.Applicants","Post.UTME.Performance","Graduate.Output","ANN-Output"))
#Store the fitted result output of the ANN model
Out<cbind(Student.enrollment_model$covariate, Student.enrollment_model$net.result[[1]])
#renaming model variables
dimnames(out) <-VV
#forecasting
lm.predict<- predict(lm.fit)
#rescale before forecasting
```

```
renormalize<- function(x,y) {return((y+min(x)) * (max(x)+min(x)))}
lm.predict1<- data.frame(renormalize(lm.predict,No.stdnt.Enrolled$No.enrolled))
stdnt.renorm<- data.frame(renormalize(Student.enrolled_norm,No.stdnt.Enrolled))
nnet.predict<- data.frame(Student.enrollment_model$net.result)
nnet.predict1<- data.frame(renormalize(nnet.predict,No.stdnt.Enrolled$No.enrolled))
nnet.predict1 <- data. Frame(nnet.predict1,No.stdnt.Enrolled$No.enrolled.
```