Rohrer's index Prediction using neural network

[*] Olaiju O.A

olaiju@yahoo.com

[-] Are S.O

oloruntoba22@yahoo.com

[+] Ojuawo Olutayo Oyewole,

teemana2000@yahoo.com

[+] Department of Computer Science, Federal Polytechnic Ilaro

[-*] Department of Mathematics and Statistics, Federal Polytechnic Ilaro

ABSTRACT

Artificial neural networks can be considered effective in making Predictions in which traditional methods and statistics are not suitable. In this article, by using two-layer feedforward network with tan-sigmoid transmission function in input and output layers, we can anticipate the prediction of Rohrer Index an anthropometric statistic which combines the height and weight of an individual into a singular metric used to classify individuals into the following categories: severely underweight, underweight, normal, overweight, and obese. We compare different artificial neural networks architectures with the traditional multiple Regression model. , Ideally the mean error would be zero and the standard deviation would be as small as possible in other to pick the best model. All of the models' means are relatively close to zero. However, the breakout occurs with standard deviation. The larger the standard deviation the greater the range of error, so ANN10 model perfumed best in predicting the Rohrer Index.

Key words: **Rohrer's Index, Artificial Neural Network, Regression model**

INTRODUCTION

Rohrer's Index is an anthropometric statistic which combines the height and weight of an individual into a singular metric. The Rohrer's Index and the Body Mass Index (BMI) serve a similar purpose in that both measures can be used to classify individuals into the following categories: severely underweight, underweight, normal, overweight, and obese. The BMI measurement assumes the body is a two-dimensional square sheet and it measures weight per square unit of area, whereas the Rohrer's Index assumes the body is a three-dimensional cube and measures weight per cubic unit of volume. The Rohrer's Index therefore takes into consideration one's width and girth unlike the BMI measurement, and assumes that width and girth are proportional to one's height. The index is identical with the Ponderal Index.

Calculating Rohrer's Index

The formula for calculating the Rohrer's Index is:

In metric units: $\frac{\text{Body weight(g)}\times 100}{(\text{Height(cm)})^3}$

In Imperial units: $\frac{\text{Body weight(lb)}\times 2768}{(\text{Height(in)})^3}$

Uses of Rohrer's Index

Once calculated the Rohrer's Index measurement can be used for many purposes. One recently investigated usage of the Rohrer's Index is medical underwriting. Medical underwriters typically use height and weight or BMI as a component in the determination of an individual's health status because one's build can be correlated to specific chronic health conditions such as diabetes and heart disease. Rohrer's Index is thought of by some to be preferable to BMI because the threshold values used in the risk stratification of individuals during the underwriting process are more consistent than those of BMI, allowing for the creation of improved and refined predictive health cost models.

In this paper we compare the ability of neural network to predict accurately the Rohrer's index against that of the traditional Multi linear regression.

Artificial Neural Networks (ANN)

ANNs are based on the present understanding of the biological nervous systems. An ANN is a massively parallel-distributed information processing system that has certain performance characteristics resembling biological neural networks of the human brain (Haykin 1994). The network consists of layers of parallel processing elements, called neurons. In most networks, the input layer receives the input variables for the problem at hand. This consists of all quantities that can influence the output. The output layer consists of values predicted by the network and thus represents model output. Between the input layer and output layer there may be one or more hidden layer. The neurons in each layer are connected to the neurons in a proceeding layer by a weight, w, which can be adjusted during training. The networks are organized by training methods, which greatly simplify the development of specific applications. Classical logic in ordinary artificial intelligence (AI) systems is replaced by vague conclusions and associative recall. This is a big advantage in all situations where no clear set of logical rules can be given. Figure 1 illustrates a three-layer neural network consisting of four neurons in input layer, four neurons in hidden layer and two neurons in output layer, with the interconnection weights between layers of neurons.

ANNs Training:

Training network is a procedure during which ANN processes a training set (inputoutput data pairs) repeatedly, changing the values of its weights, according to a predetermined algorithm, to improve its performance.

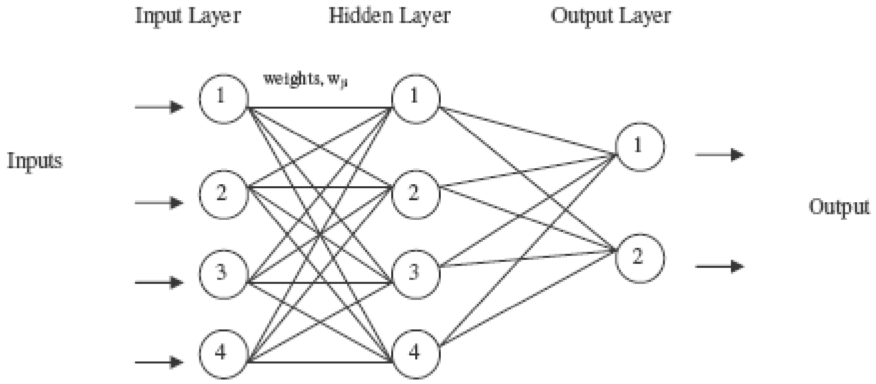

Figure 1 Configuration of Three-layer Neural Network

Back-propagation is perhaps the most popular algorithm for training ANNs. The back-propagation algorithm gives a prescription for changing the weights, w_{ji}, in any feedforward network to learn a training vector of input-output pairs. It is a supervised learning method in which an output error is fed back through the network, altering connection weights so as to minimize the error between the network output and the target output. The following equation is used for the connection weights adjustment.

$$\Delta W_{ij}(n) = -\ \varepsilon * \left(\frac{\partial E}{\partial W_{ij}}\right) + \ \alpha * \ \Delta W_{ij}(n-1) \qquad (1)$$

Where

$\Delta w_{ij}(n)$ and $\Delta w_{ij}(n-1)$ = weight increments between node i and j during the nth and (n-1)th pass, or epoch.

ε = learning rate.

α = momentum.

The momentum factor can speed up training in very flat regions of the error surface and help prevent oscillations in the weights. A learning rate is used to increase the chance of avoiding the training process being trapped in local minima instead of global minima. Back-propagation is a first-order method based on the steepest gradient decent, with the direction vector being set equal to the negative of the gradient vector. Consequently, the convergence may progress slowly and may show oscillatory behavior. It is also possible for the training process to be trapped in the local minimum despite the use of learning rate. The network architecture is required to be prefixed by trials to do so

Advantages of ANN Models

ANNs offer valuable characteristics unavailable together elsewhere (Zealand et al. 1999): ANN models infer solutions from data without prior knowledge of the regularities in the data; they extract the regularities empirically. ANN networks learn the similarities among patterns directly from instances or examples of them. ANNs can modify their behavior in response to the environment (i.e. shown a set of inputs with corresponding desired outputs, they self adjust to produce consistent responses). ANNs can generalize from previous examples to new ones. Generalization is useful because real-world data are noisy, distorted, and often incomplete. ANNs are also very good at the abstraction of essential characteristics from inputs containing irrelevant data. ANN models are non-linear; that is, they can solve complex problems more accurately than linear techniques do. ANN models can provide predications of output parameters in real time in response to simultaneous and independent fluctuations of the values of model input parameters. Finally, ANNs are highly parallel. They contain many identical, independent operations that can be executed simultaneously, often making them faster than alternative methods.

Development of ANN Models

Selection of Model Inputs and Outputs The selection of appropriate input variables for proper mapping desired output variables is a very important step to ensure successful application of ANN models in hydrologic processes. Normally, it starts with a set of inputs that are KNOWN to affect the process, then add other inputs that are suspected of having a relationship in the process one at a time. A good understanding of the hydrologic system can lead to better choice of input variables for proper mapping. This will help in avoiding loss of information that may result if key variables are omitted, and also prevent inclusion of spurious inputs that tend to confuse the training process. A sensitivity analysis can be used to determine the relative importance of a variable (Maier and Dandy 1996) when sufficient data is available. The input variables that do not have a significant effect on the performance of an ANN can be trimmed from the input vector, resulting a more compact network. (Liong, Lim and Paudyal 2000).

Hidden Layer Size The number of neurons in the input and output layers is determined by the number of input and output variables for a given system. The size of a hidden layer is one of the most important considerations when solving actual problems using multilayer feedforward networks. If there are fewer hidden layer neurons, there may not be enough opportunity for the neural network to capture the intricate relationships between indicator parameters and the nature of contaminating source. Too many hidden layer neurons not only require a large computational time for accurate training, but may also result in 'overtraining' (Brion et. al, 1999). A neural network is said to be 'overtrained' when the network focuses on the characteristics of individual data points rather than just capturing the general patterns present in the entire training set.

The following function can be recommended calculating the number of neurons in
hidden layer: (http://www.neuralware.com/frequent.htm)

$$\mathrm{N} \; = \; (\text{Number of input} \; + \; \text{output}) * \left(\frac{2}{3}\right) \qquad (2)$$

Where

N = the number of neurons in hidden layer.

Data Initialization

The contribution of an input/output will depend heavily on its variability relative to other inputs/outputs. If one input/output has a range of 0 to 1, while another input/output has a range of 0 to 1,000,000, then the contribution of the first input/output to the distance will be swamped by the second input/output. So it is essential to rescale the inputs/outputs so that their variability reflects their importance, or at least is not in inverse relation to their importance. One of the most useful ways to standardize inputs is mean 0 and standard deviation 1 method, which is shown as following: N

$$\mathrm{mean_i} = \frac{\sum \mathrm{X_i}}{\mathrm{N}} \qquad (3)$$

$$\text{std} = \sqrt{\frac{\sum(X_i^2 - \text{mean}_i)}{N - 1}} \qquad (4)$$

$$S_i = \frac{X_i - \text{mean}_i}{\text{Std}} \qquad (5)$$

Xi = value of the raw input/output variable X for the ith training case.

Si = standardized value corresponding to Xi.

N = number of training cases

Once the output values are obtained from an ANN model, the actual values are obtained by the inverse transformation of the equation:

$$T = 0 \times \text{Std} + \text{mean} \qquad (6)$$

T = Actual Output Value

O = output from ANN model.

Some Other Inspects

Initialization of weights and threshold values is an important consideration. The closer the initial guess is to the optimum weight space, the faster the training process. However, there is no way of making a good initial guess of weights, and they are initialized in a random fashion. Learning rate affect the speed of convergence. If it is large, the weights will be changed more drastically, but this may cause the optimum to be overshot. If it is small, the weights will be changed in smaller increments, thus causing the system to converge more slowly with little oscillation. Normally it is in the range of 0.2 to 0.5.

Software Support

The software used for this study are Matlab (version 7.6.0(R2008a)). Matlab is developed by The Math Works, Inc.. and SPSS v16

.

Several authors have done comparison studies between ANNs against traditional linear and non-linear regression methods in many field of studies (Anmala et al. 2000, Tokar and Johnson 1999, Zhang et al. 2000, Elshorbagy et al 2000). The results showed that the artificial neural network model generally performs better than other regression models in terms of accuracy and consistency.

Development of Neural Network Models

Neural network models are developed through the following steps:

1. Complete historical data analysis and literature reviews to establish the air quality and precipitation chemistry phenomena that influence acid rain conditions.

2. Select parameters that accurately represent these phenomena and are readily available on a forecast basis.

3. Identify the most significant variables based on mass balance and statistical analysis techniques.

4. Create three data sets: 1) a training data set to train the network, 2) a verification data set to determine when the network's general performance is maximized through early stopping, and 3) a testing data set to evaluate the generalization ability of the trained network. The developmental data sets should contain enough data.

5. Train the data using neural network models and discover the most appropriate network architecture for the problem.

6. Test the generally trained network on a test data set to evaluate the performance. If the results are satisfactory, the network is ready to be used for forecasting

The neural network models were developed using MatLab. 100 data sets were available for the network Data were separated into three groups: training, verification and testing (www.statsoft.com/textbook/stneunet.html).

The 100 data sets were randomly divided into three groups: 70 as training sets, 15 as verification sets and 15 as testing set. The training set was used to develop the neural network. The verification set was employed to determine when the network's general performance was maximized through early stopping. And the testing data set was used to evaluate the generalization ability of the trained network (U.S.EPA, 1999). In the training process, small weights were assigned randomly to the connections between neurons. Then the weights, and biases were modified until the error between the predicted data and the observed data was minimized based on the topology of the ANN and the learning technique. It is desired that the difference between the predicted and the observed values in the output vector be as small as possible. In the testing process, the network was tested for its generalization ability with the observed output after the training process was completed. When the neural networks are tested successfully, they can be used for prediction.

Neural networks are sensitive to the number of neurons in the hidden layers. Insufficient neurons can lead to underfitting. Too many neurons can contribute to overfitting, in which all training points are well fit, but the fitting curve oscillates wildly between these points (Neural Network Toolbox User's Guide 5-72). To obtain the best fit to the given data, various neural network architectures were attempted to obtain optimal models for predicting RI as a function of Weight and Height. The feedforward backpropagation (BP) algorithm was applied to all the neural network development in this study. The BP is an approximate steepest descent algorithm where a mean square error serves as the performance index. It is widely employed in all areas of ANN application.

In the neural network development, different scenarios on the number of hidden layers, the number of neurons in each layer, and the type of transfer function for each neuron were analyzed. Then the trained networks were tested using the testing data sets and the minimum squared error (MSE) method by modifying the network weights. It has been noticed that networks with two hidden layers of neurons may tend to remember the training data instead of generalizing it into patterns (Grubert, 2003). so, one hidden layer networks were tried. By increasing the amount of neurons in the hidden layer, the training objective was achieved successfully.

The best network for the RI had one hidden layer with 10 neurons.. The transfer functions were sigmoidal for the hidden neurons, and linear for the input and output neurons. With these transfer functions, a three-layer network can approximate any function with arbitrary accuracy.

Multiple Linear Regression

A more traditional statistical forecasting tool is regression analysis. This method uses the sum of the least squared errors to fit a curve to a data set. We predicted the RI values using the same data used in the Neural Networks. The dependant variable was designated as the R and the independent variables are Weight(kg) and height(m)

Using the data analysis tool in SPSSv16, a multiple linear regression analysis was performed on the data set and the equation is:

$$\mathbf{Rohrer's\ index = 40.136 + 0.240\big(BODYWEIGHT(KG)\big) - 24.660(HEIGHT(M)}$$

Data Presentation

ANN20	performance	regression
train	3.690243132567453e-09	0.999999999868584
validation	2.765475512676415e-05	0.999999203046382
.test	1.874043346279464e-05	0.999999623384292
Epoch	1000	

Table 1: ANN20 Performance

ANN15	performance	regression
train	9.340705086931404e-09	0.999999999701148
validation	2.201611379627460e-05	0.999999294451525
.test	4.969942057490335e-04	0.999983381850686
Epoch	1000	

Table 2: ANN15 Performance

ANN10	performance	regression
train	2.213601131531583e-09	0.999999999929355
validation	3.805841827053738e-05	0.999998881356107
.test	1.379864877546815e-08	0.999999999566209
Epoch	1000	

Table 3: ANN10 Performance

ANN5	performance	regression
train	7.424439674493788e-08	0.999999997372040

validation	1.423893742108638e-07	0.999999990552112
.test	1.002971281270289e-04	0.999999222298562
Epoch	1000	

Table 4: ANN5 Performance

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Change Statistics					Durbin-Watson
					R Square Change	F Change	df1	df2	Sig. F Change	
1	.984[a]	.969	.968	.70714	.969	1829.325	2	117	.000	2.061

a. Predictors: (Constant), HEIGHT(M), BODYWEIGHT(KG)

b. Dependent Variable: RI

Table 5: Multiple Regression Model Summary

ANN20(RI)PRE	ANN20(RI)PRE	ANN20(RI)PRE	ANN20(RI)PRE	MR(RI)PRE	RI
15.3942	15.3937	15.3938	15.3937	15.6004	15.3938
8.9208	8.9165	8.9164	8.916	8.2093	8.9163
10.3822	10.3805	10.3806	10.3809	10.4356	10.3806
8.7129	8.7182	8.7179	8.7176	7.9766	8.7179

10.7856	10.7816	10.782	10.7828	10.8315	10.782
21.4912	21.4658	21.4932	21.4929	20.633	21.4932
15.3768	15.3764	15.3767	15.3765	15.5935	15.3767
18.3722	18.3793	18.3704	18.3705	18.0039	18.3704
12.2117	12.213	12.2124	12.2128	12.5925	12.2125
10.2789	10.2847	10.2868	10.2875	10.1334	10.2871
10.6062	10.6062	10.6062	10.6074	10.5849	10.6062
10.8754	10.8768	10.8768	10.8771	11.0417	10.8768
9.4153	9.4153	9.4153	9.4151	8.9352	9.4153
7.5257	7.5523	7.554	7.5553	5.7502	7.5541
12.4662	12.4659	12.466	12.4662	12.8878	12.4661
17.032	17.0315	17.0317	17.0318	17.0244	17.0316
12.3505	12.3477	12.3457	12.3454	13.0022	12.3457
19.4026	17.2822	17.2614	17.2622	18.5358	17.2603
9.8843	9.8885	9.8893	9.8892	9.6819	9.8892
14.736	14.7362	14.736	14.736	15.2703	14.7361

Table 6: ANNs and Multiple regression (RI) Predictions

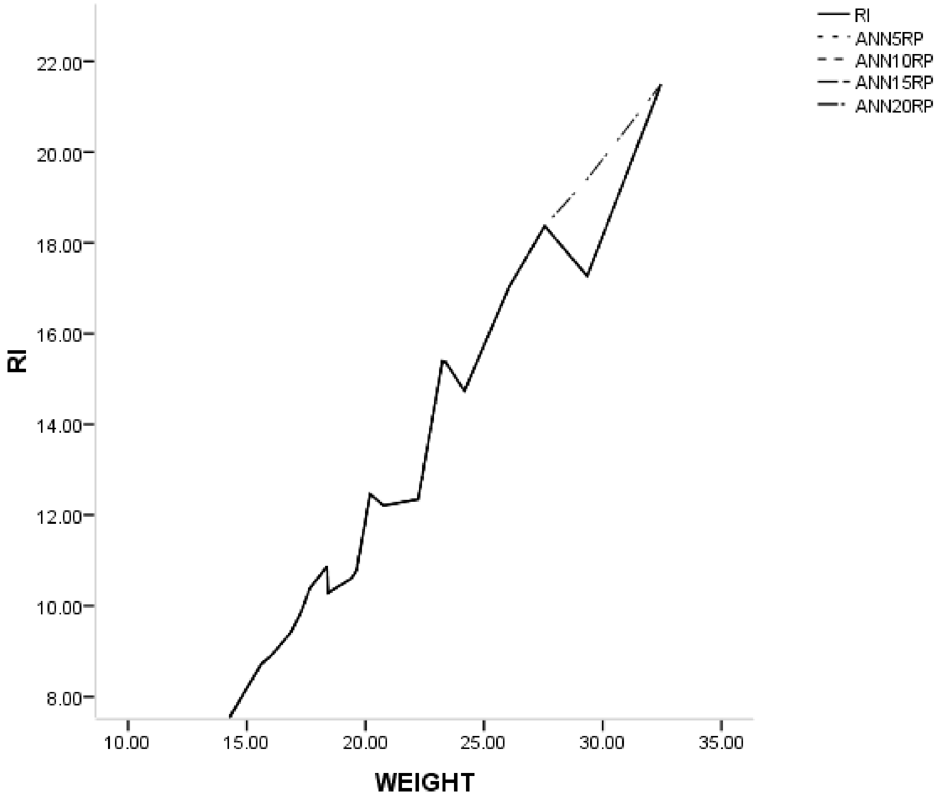

Figure 2: Graphs of the ANNs and MR predictions

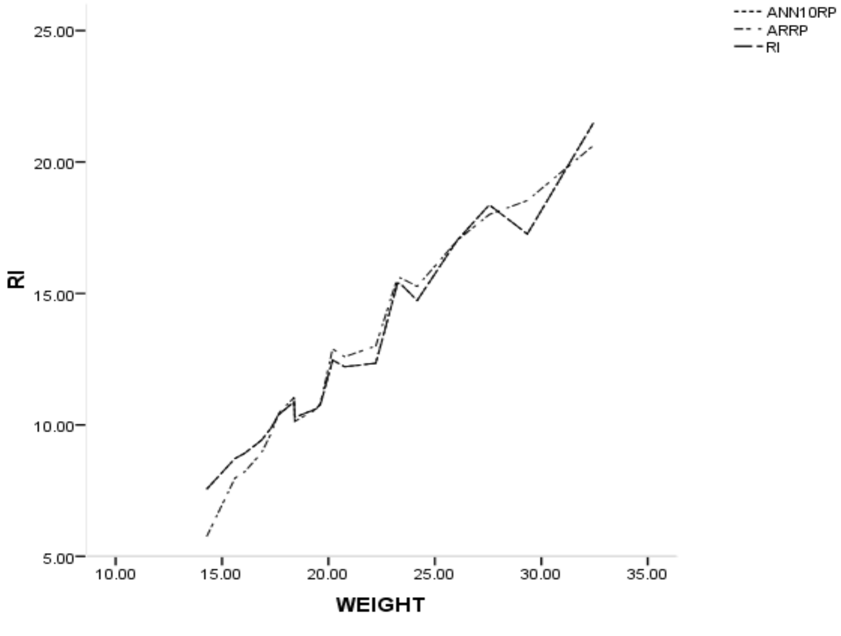

Figure3: Graphs of ANN10, Multiple Regression and actual RI

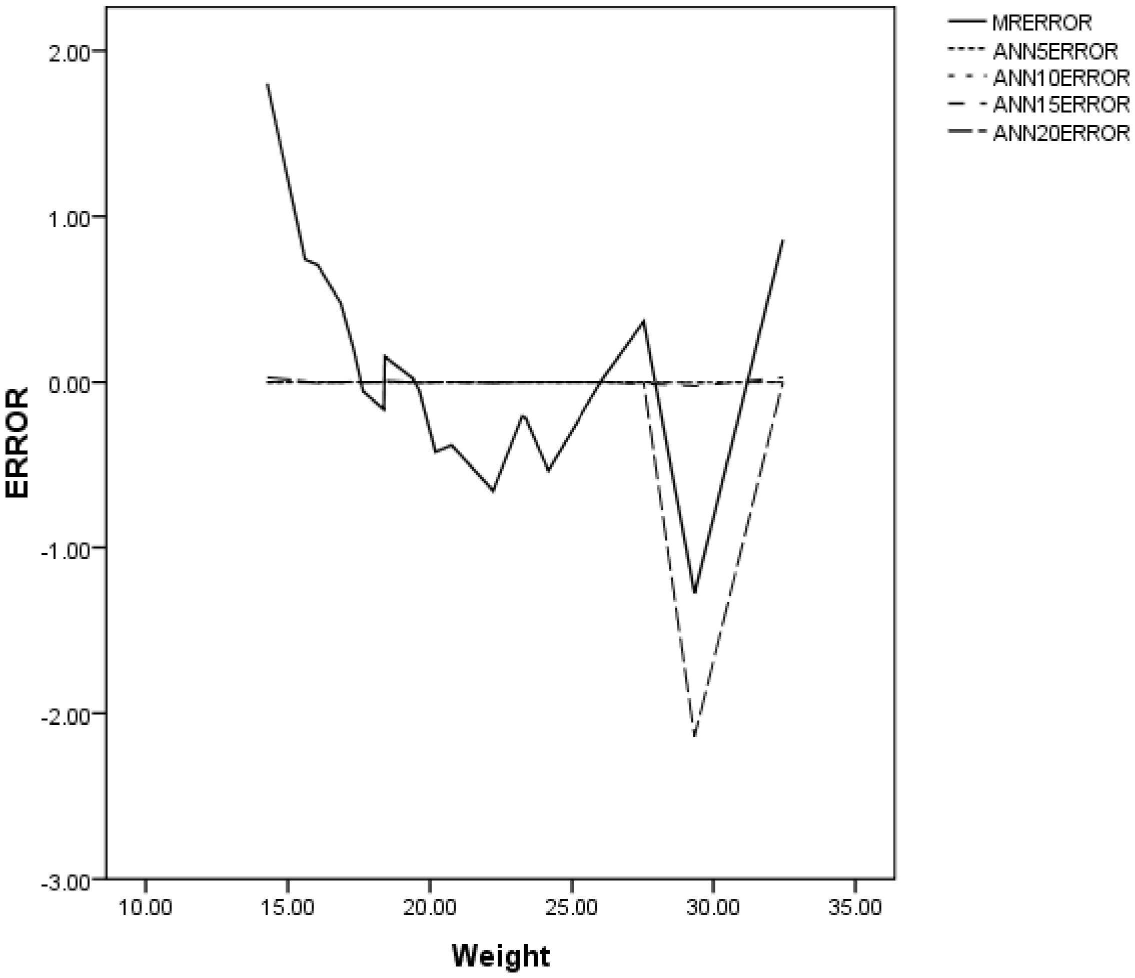

Figure 4:Graps of Errors of all the Models

Models	R-Squre
ANN20	0.999999999868584
ANN15	0.999999999701148
ANN10	0.999999999929355
ANN5	0.999999997372040
MR	0.969

Table 7: R-squares of all models

The R square value represents the proportion of variation in the dependant variable that is explained by the independent variables. The better the model explains variation in the dependant variable, the higher the R square value. Without further comparison, the ANN models best explain variation in the dependant variable, followed by the Regression Model. In examining Figure , RI vs. All Model Predictions, it is relatively easy to visually verify that the network models perform better than the regression model. This differs from the model ranking due to R square values.

In Table 8 , the ranked error statistics are provided for comparison. These statistics are all based on RI error.

Models	Means	Std
ANN20	-0.1054	0.4795
ANN15	-2.0000e-005	0.0083
ANN10	-3.5000e-005	2.6611e-004
ANN5	-2.5000e-004	5.9956e-004
MR	0.0694	0.6556

Table 8: Means and Std of all models

Ideally, the mean error would be zero and the standard deviation would be as small as possible. All of the models' means are relatively close to zero. However, the breakout occurs with standard deviation. The larger the standard deviation the greater the range of error, so ANN10 Network is more accurate.

Conclusion

ANNs and statistical methods have similarities in many aspects. Both approaches are used to model a relationship between the dependent and independent variables. It has been observed that any generalized linear model can be mapped onto an equivalent single-layer neural network (Warner and Misra, 1996). For example, given the linear equation $= \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n$, the independent variable (x_i) correspond to the inputs of the neural network and the response variable y to the output. The coefficients, β_i , correspond to the weights in the neural network. There are differences between ANNs and statistical regression models. In regression models a functional form is imposed on the data. In the case of multiple linear regression this assumption is that the outcome is related to a linear combination of the independent variable. If this assumed model is not correct, it will lead to error in the prediction. So traditional linear models are simply inadequate when it comes to modeling data that contains non-linear characteristics. While ANN models do not assume any functional relationship and let the data define the functional form. Thus ANNs is extremely useful when there is no idea of the functional relationship between the dependent and independent variables.

Recommendations

The potential of artificial neural network methodology has been highlighted for successfully tackling the realistic situation in which exact nonlinear functional relationship between response variable and a set of predictors is not known.

Although ANNs may not be able to provide the same level of insight as many statistical models do, it is not correct to treat them as “black boxes”. In fact, one active area of research in ANN is ‘understanding the effect of predictors on response variable’. It is hoped that, in future, research workers would start applying some of the other more advanced ANN models, like ‘Radial basis function neural network’, and ‘Generalized regression neural network’ in their studies.

Reference

Anmala, J., Zhang, B., and Govindaraju, R.S (2000). "Comparison of ANNs and Empirical Approaches for Predicting Watershed Runoff" Journal of Water Resources Planning and Management, Vol. 126, No.3 May/June, 2000, page 156-166.

Atiya, A.F, El-Shoura, S. M, Shaheen, S.I., and El-Sherif, M. S (1999). "A comparison Between Neural-Network Forecasting Techniques – Case Study: River Flow Forecasting" IEEE Transactions on Neural Network, Vol. 10, No. 2, March 1999.

Brion, Gail Montgomery and Lingireddy, Srinivasa (1999). "A Neural Network Approach to Identifying Non-Point Sources of Microbial Contamination." Water Resources. Vol. 33 No. 14 pp. 3099-3106. 1999.

Elshorbagy A, Simonovic S.P and Panu U.S. (2000). "Performance Evaluation of Artificial Neural Networks for Runoff Predication" Journal of Hydrologic Engineering. Vol.5, No. 4, October, 2000. 424-427.

Haykin, S. (1994). Neural networks: a comprehensive foundation. Mac-Millan. New York.

Hsu, K., Gupta, H.V., and Sorooshian, S. (1995). "Artificial neural network modeling of the rainfall-runoff process." Water Resources Research, vol. 31, No. 10, Page 2517-2530.

Kuligowski, R.J., and Barros A.P (1998). "Experiments in short-term precipitation forecasting using artificial neural networks." Mon. Wea. Rev., 126, 470-482.

Liong, Shie-yui, Lim, W. and Paudyal, G. N., (2000). "River stage forecasting in Bangladesh: Neural network approach." Journal of Computing in Civil Engineering. Vol. 14, No. 1. January, 2000. Page 1-8.

Maier, H. R., and Dandy, G. C (1996). "The use of artificial neural networks for the prediction of water quality parameters." Water Resources Research, vol. 32, No. 4, 1013-1022 pp.

Rogers, L.L. and Dowla, F.U. (1994). "Optimization of groundwater remediation using artificial neural networks with parallel solute transport modeling." Water Resources Research. 30, 2: 457-481.

Sajikumar, N., and Thandaveswara, B.S. (1999). "A non-linear rainfall-runoff model using an artificial neural network." Journal of Hydrology, 216 (1999) 32-55.

Smith, J. and Eli, R.N. (1995). "Neural-network models of rainfall-runoff process." Journal of Water Resources Planning and Management, Vol. 121, No.6 November/December, 1995, page 499-508.

Tokar, A.S, and Johnson, P.A (1999). "Rainfall-Runoff Modeling Using Artificial Neural Networks". Journal of Hydrologic Engineering. Vol.4, No. 3, July, 1999. 232-239.
Warner, B., and Misra, M. (1996). "Understanding Neural Networks aas Statistical Tools" The American Statistician, 50, 284-293.

Zealand, C.M., Burn, D.H., and Simonovic S.P. (1999). "Short term streamflow forecasting using artificial neural