Chapter 15

ARTIFICIAL NEURAL NETWORK FOR TIME SERIES MODELLING

Mrinmoy Ray, K. N. Singh, Kanchan Sinha and Shivaswamy G. P.

INTRODUCTION

An artificial neural network (ANN), otherwise called neural network (NN), is one of the popular machine learning techniques that is inspired by the structure as well as functional aspects of biological neural networks the human brain especially. A neural network made out of various interconnected simple processing elements called neurons or nodes. Each node receives an input signal which is the aggregate "information" from other nodes or external stimuli, processes it locally through an activation or transfer function and produces a transformed output signal to other nodes or external outputs. This information processing characteristic makes ANNs an effective computational device and able to learn from examples and is then to generalize to examples never seen before.

A Time series (TS) is an ordered sequence of observations of a variable at equally spaced time intervals (monthly price data of a commodity, yearly crop yield and daily temperature data etc.). The motivation behind the use of time series model is to predict future values based on previously observed values. The most popular time series model is Auto Regressive Integrated Moving Average (ARIMA) (Box *et al.,* 2009 and Makridakis *et al.,* 1989). However, the main drawback of ARIMA model is that it can only deal with linear time series data (Ray *et al.,* 2016). Even though there are divergent statistical models to deal with non-linear time series data such as bilinear, Autoregressive Conditional Heteroscedasticity (ARCH) model, Generalized Autoregressive Conditional Heteroscedasticity (GARCH) model and threshold autoregressive (TAR) model etc. But these models have some specific assumption which is not always plausible to identify while dealing with real world time series data (Jha and Sinha, 2014). Therefore, in the gamut of time series modeling Artificial Neural Network (ANN) (Zhang *et al.,* 1998; Remus and O' Connor, 2001 and Mukerjee *et al.,* 2016) has notched up much popularity in modeling non-linear dynamics and subsequently rendering the non-linear forecasting. The main feature of this approach is that it doesn't require prior assumption of the time series data under consideration, rather it is to a great extent depends upon pattern of the data popularly known as data-driven approach.

ANN Architecture

In general, an ANN can be partitioned into three sections, named layers, which are discussed as

Input layers

These layers are responsible for receiving information (data), signals, features, or measurements from the external environment. These inputs (samples or patterns) are usually normalized within the limit values produced by activation functions. This normalization results in better numerical precision for the mathematical operations performed by the network.

Hidden, intermediate, or invisible layers

These layers are composed of neurons which are responsible for extracting patterns associated with the process or system being analyzed. These layers perform most of the internal processing from a network.

Output layers

These layers are also composed of neurons, and thus are responsible for producing and presenting the final network outputs, which result from the processing performed by the neurons in the previous layers.

ANN approach to time series forecasting

In the domain of time series analysis, the inputs are typically the past observations series and the output is the future value. The ANN performs the following nonlinear function mapping between the input and output

$$y_t = f(y_{t-1} + y_{t-2}, \ldots, y_{t-p}, w) + \varepsilon_t$$

where, w is a vector of all parameters and f is a function of network structure and connection weights. Therefore, the neural network resembles a nonlinear autoregressive model.

Single hidden layer multilayer feed forward network is the most popular for time series modeling and forecasting. This model is characterized by a network of three layers of simple processing units. The first layer is input layer, the middle layer is the hidden layer and the last layer is output layer (Fig 1).

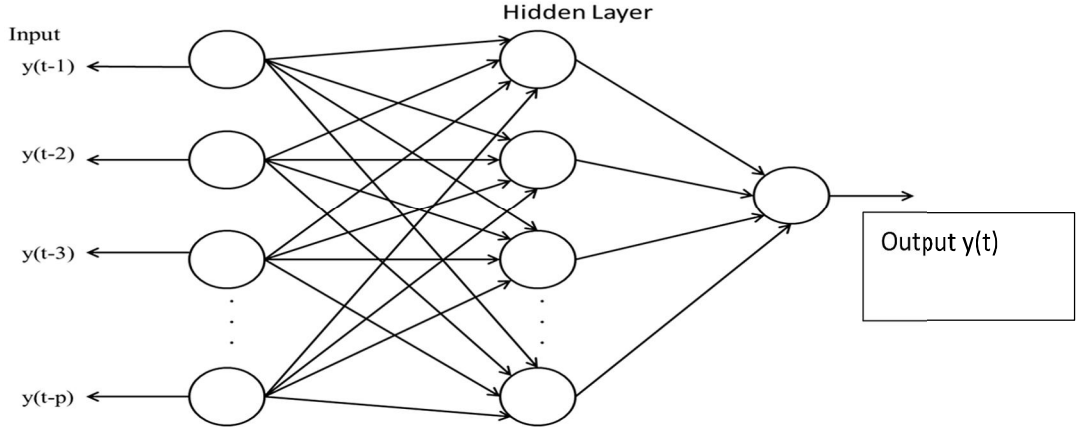

Fig 1: Architecture of ANN for time series forecasting

The relationship between the output (y_t) and the inputs (y_{t-1}, y_{t-2},...,y_{t-p}) can be mathematically represented as follows:

$$y_t = f\left(\sum_{j=0}^{q} \omega_j g\left(\sum_{i=0}^{p} \omega_{ij} y_{t-i}\right)\right)$$

where, $\omega_j (j = 0,1,2,\ldots\ldots,q)$ and $\omega_{ij} (i = 0,1,2,\ldots\ldots,p,\ j = 0,1,2,\ldots\ldots,q)$ model parameters often called the connection weights, p number of input nodes q number of hidden nodes, g and f activation function at hidden and output layer respectively. Activation function defines the relationship between inputs and outputs of a network in terms of degree of the non-linearity.

For time series forecasting sigmoid activation function is employed in hidden layer and identity activation function is employed in the output layer which are given in Table 1.

Table 1: Time series forecasting sigmoid activation function

Activation function	Equation
Identity	x
Sigmoid	$\frac{1}{1+e^{-x}}$

The selection of appropriate number of hidden nodes as well as optimum number of lagged observation p for input vector is important in ANN modeling for determination of the autocorrelation structure present in a time series. Though there are no established theories available for the selection of p and q, hence experiments are often conducted for the determination of the optimal values of p and q. The connection weights of ANNs are determined by Gradient decent back propagation algorithm which is described here.

The objective of training is to minimize the error function that measures the misfit between predicted value and actual value. The error function which is widely used is mean squared error which can be written as:

$$E = \frac{1}{N}\sum_{n=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{n=1}^{N}\left\{y_t - f\left(\sum_{j=0}^{q} \omega_j g\left(\sum_{i=0}^{p} \omega_{ij} y_{t-i}\right)\right)\right\}^2$$

where N total number of error terms. The parameters of the neural network are ω_j and ω_{ij} estimated by iteration. Initial connection weights are taken randomly from uniform distribution. In each iteration the connection weights changed by an amount $\Delta\omega_j$

$$\Delta\omega_j(t) = -\eta \frac{\partial E}{\partial \omega_j} + \delta \Delta\omega_j(t-1)$$

where, η , learning rate and $\frac{\partial E}{\partial \omega_j}$, partial derivative of the function E with respect to the weight ω_j ; δ , momentum rate. The $\frac{\partial E}{\partial \omega_j}$ can be represented as follows-

$$\frac{\partial E}{\partial w_j} = -e_j(n) \times f'(x) \times y_j(n)$$

where $e_j(n)$, residual at nth iteration

$f'(x)$ = derivative of the activation function in the output layer. As in time series forecasting the activation function in the output layer is identity function hence $f'(x)$ =1. $y_j(n)$ is the desired output. Now connection weights in from input to hidden nodes changed by an amount $\Delta\omega_{ij}$

$$\Delta\omega_{ij}(t) = -\eta \frac{\partial E}{\partial \omega_{ij}} + \delta \Delta\omega_{ij}(t-1)$$

where

$$\frac{\partial E}{\partial w_{ij}} = g'(x) \times \sum_{j=0}^{q} e_j(n) * w_j(n)$$

where $g'(x)$ is the activation function in the hidden layer. For sigmoid activation function

$$g'(x) = \frac{\exp(-x)}{(1+\exp(-x))^2}$$

Learning rate is user defined parameter known as tuning parameter of neural network which determine how slow or fast the optimal weight is obtained. The learning rate must be set small enough to avoid divergence. The momentum term prevents the learning process from setting in a local minimum. Though there are no established theories available for the selection of learning rate and momentum, hence experiments are often conducted for the determination of the learning rate and momentum.

Step by step modeling procedure

Division of the data

Data divided into training and test sets. The training sample is used for ANN for model development and the test sample is utilized to evaluate the forecasting performance. Sometimes a third one called the validation sample is also utilized to avoid the overfitting problem or to determine the stopping point of the training process. It is common to use one test set for both validation and testing purposes particularly for small data sets.

The literature suggests little guidance in selecting the training and testing sets. Most commonly used rule are 90% vs. 10%, 80% vs. 20% or 70% vs. 30%, etc.

Data normalization

Nonlinear activation functions such as the sigmoid function typically have the squashing role in restricting the possible output from a node to, typically, (0, 1). Hence, data normalization is done prior to training process begins.

Normalization procedure

Linear transformation to [0,1]: $X_n=(X_0-X_{min})/ (X_{max}-X_{min})$

Statistical normalization: $X_n=(X_0-\text{mean}(X))/\text{var}(X)$

Simple normalization: $X_n=X_0/ X_{max}$

Selection of appropriate number of hidden nodes as well as optimum number of lagged

There are no established theories available for the selection of p and q, hence experiments are often conducted for the determination of the optimal values of p and q.

Estimation of connection weights

Estimation of connection weights are determined by learning algorithm. For time series forecasting most commonly used learning approach is gradient decent back propagation algorithm.

Evaluating forecasting performance

Forecasting performance can be computed by several approaches. Some of the approaches are given here

$$MAPE=\frac{1}{n}\sum_{t=1}^{n}\left|y_t-\hat{y}_t\right|/\,y_t\times 100$$

$$MSE=\frac{1}{n}\sum_{t=1}^{n}\left(y_t-\hat{y}_t\right)^2$$

where, n, total number of forecast values; y_t, actual value at period t and , corresponding forecast value. The model with less MAPE/MSE is preferred for forecasting purposes.

ILLUSTRATION

Monthly data on wholesale price of rice (January, 2010 to October, 2018) of all-India level data have been utilized for Illustration purpose. The plot of considered time series data is given here (Fig 2).

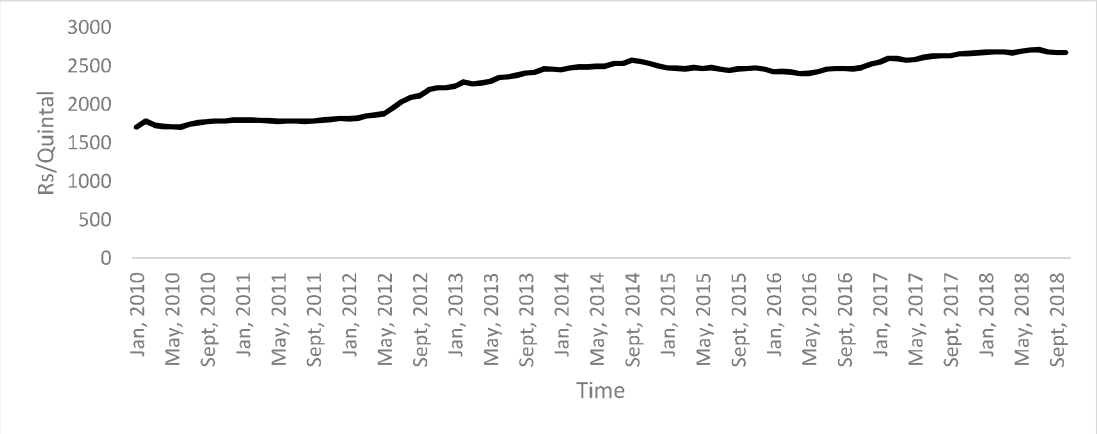

Fig 2: Wholesale price of rice (January, 2010 to October, 2018) of all-India level

Data from January, 2010 to November, 2017 (90%) were used for model construction. The summary of the fitted neural network model is given in Table 2. The ANN was fitted employing "forecast" package of R software. The R code is given in Annexure.

Table 2: Summary of fitted ANN

Parameters	**ANN**
Number of input (lag)	1
Number of hidden unit	2
Activation function in hidden unit	Sigmoid
Activation function in output unit	linear
Learning algorithm	Gradient decent back propagation

From December, 2018 to October, 2018 (10%) were used to check the forecasting performance. The Actual versus ANN model values plot is given in Fig 3.

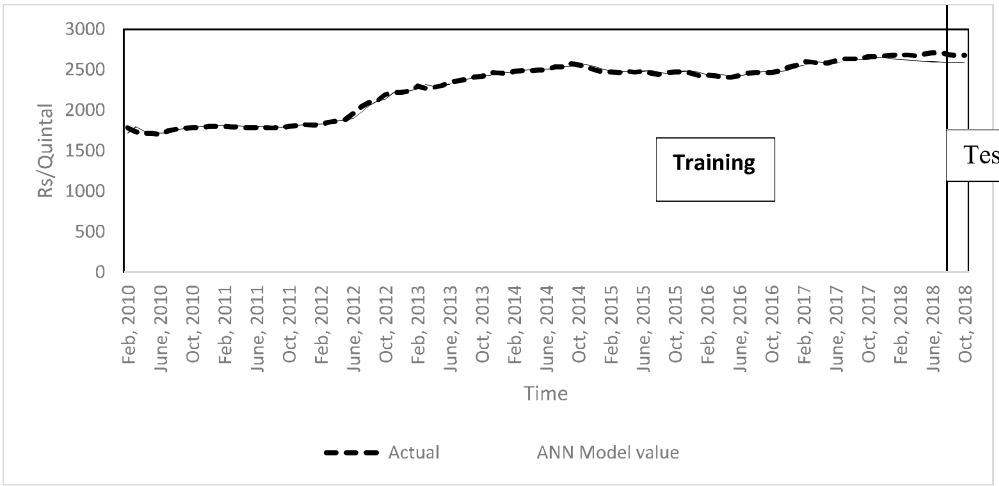

Fig 3: The actual versus ANN model values plot

Fig 3 reveals that ANN model is good fit for the considered time series data. The MAPE and MSE have been computed for training as well as testing set Table 3.

Table 3: MAPE and MSE values of fitted ANN model

Model	MAPE		MSE	
	Training	Testing	Training	Testing
ANN	0.81	2.83	567.20	6584.01

Based on the values of MAPE and MSE the performance of the ANN model can be compared with other time series model like ARIMA, GARCH etc.

REFERENCES

Box, G. E. P., G. M. Jenkins and G. C. Reinsel (2009), Time Series Analysis*: Forecasting and* Control (3rd ed.), San Francisco*: Holden-Day.*

Jha, G. K. and K. Sinha (2014), Time-delay neural networks for time series prediction: an application to the monthly wholesale price of oilseeds in India. *Neural Computing and Applications*, 24 (3): 563-571.

Makridakis, S., S. C. Wheelwright and R. J. Hyndman (1998), Forecasting: Methods and Applications, 3rd Editon, Chichester: Wiley.

Mukherjee, A., S. Rakshit, A. Nag, M. Ray, H. L. Kharbikar, S. Kumari, S. Sarkar, S. Paul, S. Roy, A. Maity, V. S. Meena and R. R. Burman (2016), Climate Change Risk Perception, Adaptation and Mitigation Strategy: An Extension Outlook in Mountain Himalaya. In: Jaideep Kumar Bisht, Vijay Singh Meena, Pankaj Kumar Mishra and Arunava Pattanayak Edition. Conservation Agriculture (pp. 257-292). Singapore. Springer Singapore.

Ray, M., A. Rai, V. Ramasubramanian and K. N. Singh (2016), ARIMA-WNN hybrid model for forecasting wheat yield time series data. *Journal of the Indian Society of Agricultural Statistics*, 70(1): 63-70.

Remus, W. and M. O'Connor (2001), Neural Networks for Time-Series Forecasting, *New york,* Springer.

Zhang, G., B. E. Patuwo and M. Y. Hu (1998), Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14: 35-62.

ANNEXURE I

```
z=read.csv(file.choose() , header=TRUE) # Uploading data in R
head(z)
zz=data.frame(z)
n = nrow(zz)
size = round(0.90*n) # Dividing 90% data as training data as rest 10% as testing data
train=zz[1:size,]
test=zz[(size+1):n,]
install.packages("forecast") # installation of forecast package
library(forecast)
fit<-nnetar(train$x)
fitv=fitted(fit) # fitted values
ff<-forecast(fit, h=nrow(test))
kk2=ff$mean# Forecasted values
```