Application of Artificial neural networks in Time series forecasting

M. Narsimulu

Research Scholar, Department of Applied Statistics,
Telangana University, Nizamabad, Telangana State, India.

Abstract: The most important lagged components in time series forecasting can be found using an advanced method introduced in this article called an artificial neural network model is Long Short-Term Memory (LSTM). Additionally, this article compares the forecasting accuracy of the traditional ARIMA model utilizing time series data with the artificial neural network model is LSTM. Collected rainfall data for India from 1901 to 2015. According to our findings, the coefficient of multiple determination (R^2), mean absolute error (MAE), root mean squared error (RMSE), mean absolute percentage error (MAPE), the best predicting accuracy is offered by Long Short-Term Memory (LSTM) neural networks, which are more advanced than traditional time series approaches and the traditional ARIMA model.

Keywords: time series forecasting, artificial neural networks, ARIMA model, LSTM.

Introduction

In spite of the consensus in most fields of knowledge on the necessity to develop accurate forecasts and the identification of their corresponding benefits, there is no one model out performs in terms of forecasting accuracy (Law & Au, 1999). In this regard, one of the most widely used method in time series forecasting is the Box-Jenkins methodology (Box & Jenkins, 1976), which is built on the fit of a different type of statistical model known autoregressive integrated moving average(ARIMA). One problem that makes developing and implementing this type of time series model difficult is that the model must be formally specified and a theoretical probability distribution for data must be assumed (Hansen, McDonald, & Nelson, 1999).

In recent years, the study of artificial neural networks (ANN) has aroused great interest in fields as diverse as biotechnology, psychology, medical sciences, health care, Data science, computer science and applications. The reason behind this interest is that ANN are universal function approximations capable of mapping any linear or nonlinear complexity function (Cybenko, 1989; Funahashi, 1989; Hornik, Stinchcombe, & White, 1989; Wasserman, 1989). Because of their flexibility in function approximation, ANN are powerful methods in tasks involving estimating continuous variables, pattern classification and forecasting (Kaastra & Boyd, 1996). In the last case, neural networks offer several potential advantages over alternative methods – mainly the ARIMA time series models – when handling problems with non-normal distribution (Havier tailed) and nonlinear data (Hansen et al., 1999). The first advantage is that ANN are very versatile and do not require formal specification of the model nor acceptance of a determined probability distribution for data. As for the second advantage, Masters (1995) demonstrates that ANN are capable of tolerating the presence of chaotic components better than most alternative methods. This capacity is particularly important, as many relevant time series possess significant chaotic components.

Thus, ANN have been successfully applied to forecasting time series in diverse fields such as biology (Aznarte et al., 2007), image processing and pattern recognition (Altay & Satman, 2005; Zou, Xia,Yang,&Wang, 2007), power consumption (Pao, 2006), medicine (Haque, Hasan, & Tazawa, 2001), meteorology (Yuval, 2000; Incerti, Feoli, Salvati, Brunetti, & Giovacchini, 2007), and tourism (Palmer, Montaño, & Sesé, 2006; Aslanargun, Mammadov, Yazici, & Yolacan, 2007). The literature suggests that the application of artificial neural networks outperforms more than conventional forecasting techniques (Sirakaya, Delen, & Choi, 2005). In this sense, the ANN model most widely used in rain fall time series forecasting is the Long Short- Term memory(LSTM), which uses a specific number of lagged terms as input signals and forecasts as output signals in the time series (Bishop, 1995). Nevertheless, the scientific literature on ANN does not provide a systematic procedure for identifying the most relevant lagged terms in forecasts, as classic forecasting techniques do. Instead, relevant terms are selected arbitrarily by users, which involves a substantial investment of time and effort.

This research paper presents a procedure called time series analysis, which allows the most relevant lagged terms in time series forecasting to be identified directly and effectively when applied to an LSTM network; the second aim is to conduct a comparison of forecasting accuracy between the artificial neural network model is Long Short-Term Memory (LSTM) resulting from applying the time series analysis with the conventional methods and the classic ARIMA model.

Related works

There are a number of approaches for time series data analysis, but LSTM Neural Network (NN) application is a well-liked option for researchers to predict rainfall. The performance of the LSTM Neural Network (NN) is acknowledged as the best alternative for the researchers among the various time series techniques, such as Exponential smoothing, Generalized Regression, Moving Average, and Auto regressive Integrated Moving Average (ARIMA), among others. Researchers have worked on analyzing time series data in a variety of ways. In this regard, a model based on LSTM Neural Network was created to anticipate the amount of rain fall, and it was compared against a model based on an auto regressive integrated moving average (ARIMA). The LSTM neural Network generated a simple, reliable structure that accurately predicted annual rain fall.

Material and Methods

Data

This Research paper study was based on rainfall data of India. The data that are used in this research were obtained from the https://www.kaggle.com , in this research the variable is rainfall data. The rainfall data used is the monthly or yearly rainfall from 1901 to 2015. This data is related to time series data, then the analysis carried out is time series analysis and forecasting.

Procedure

ARIMA models

ARIMA (autoregressive integrated moving average) models were developed by Box and Jenkins (1976) and are the most popular technique is used for analyzing time series. The methodology combines up to three possible types of processes based on a time series' characteristics: The first process is autoregressive (AR), the second is series differencing, and the third is moving averages (MA). The specification of an ARIMA model generally has the following form.

$\emptyset(B)\,(1-B)^d y_t = \varphi(B)\varepsilon_t$

Where; $\emptyset(B)$, $\varphi(B)$ are the Autoregressive and Moving average polynomial as defined by:

$\emptyset(B) = 1- \emptyset_1 B - \emptyset_2 B^2 - ---------- - \emptyset_p B^p$

And

$\varphi(B) = 1- \varphi_1 B - \varphi_2 B^2 - ---------- - \varphi_q B^q$

Where $\emptyset_1$, $\emptyset_2$, ------- $\emptyset_p$ are the autoregressive coefficients while φ_1, φ_2 ----- φ_q are the moving average coefficients

In the above, d represents the order of differencing, B is the backshift operator, i.e., $By_t = y_t - 1$, p and q are the order of autoregressive and moving average respectively. ε_t is a white noise process.

In a simplified form, these kinds of models are defined by the ARIMA representation (p, d, q).

The four phases involved in building an ARIMA model are identification, estimation, validation and, lastly, prediction. The identification phase is based on observing autocorrelation functions (ACF) and partial autocorrelation functions (PACF). The estimation phase calculates the parameters needed to build the model and the validation phase tests whether the selected model fulfills all the requirements, so that it can be used correctly in the prediction phase.

Artificial Neural Network Models

Artificial neural network (ANN) models are information processing systems whose structure and functioning are inspired by biological neural networks. They have three fundamental characteristics that differentiate them from any other processing system: parallel processing, distributed memory, and adaptation to environment. ANN can be used in two ways, either as models for studying the nervous system and cognitive phenomena, or as tools for solving practical problems, such as patterns classification and functions approximation (West, Brockett, & Golden, 1997). The Long Short-Term Memory (LSTM)) (also known as a Recurrent neural network) is the preferred type of neural network model for the latter, since it is capable of solving a wide range of problems. For the purposes of this paper, the LSTM is also the neural network most often used in time series forecasting (Bishop, 1995; Kaastra & Boyd, 1996). An LSTM network is composed of an input layer, an output layer, and one or more hidden neural layers, although it has been demonstrated that a single hidden layer suffices for most problems (Funahashi, 1989; Hornik, Stinchcombe, & White, 1989) (Figure 4). From a statistical standpoint, each input neuron represents a predictor variable, whereas output neurons represent the response variables in the LSTM network. The main explanation behind the popularity of the LSTM network in the field of data analysis is that it is a universal functions approximation. This means that an LSTM network with sufficient data and hidden neurons can approximate any arbitrary function to the desired level of accuracy (Cybenko, 1989; Funahashi, 1989; Hornik et al., 1989; Wasserman, 1989). This principle turns LSTM networks into flexible, all-purpose nonlinear tools. This means LSTM networks are capable of adequately modeling trends, nonlinear relations, and chaotic components, all of which are present in most time series. Modelling univariate time series customarily uses a specific number of lagged terms in the time series as inputs and forecasts as outputs in this neural network (Bishop, 1995). The number of input neurons determines the number of prior time points that will be used in each forecast, while the number of output neurons determines the forecast horizon. One of the major drawbacks to ANN is how difficult it is to understand the nature of the internal representations generated by an LSTM network in response to a certain task (Rzempoluck, 1998). Unlike classic statistical models, the effect or relevance that each input variable (predictor) has on output variables (response) is not clearly ascertainable in networks. In their application to time series forecasting, this means that neural networks do not have a systematic procedure for identifying the most relevant lagged terms in forecasting, as classic forecasting techniques do. Instead, as mentioned above, identifying or selecting relevant terms is accomplished through an arbitrary selection by users or experimental trial and error, which involves a considerable investment of time and effort. Nevertheless, this perception of ANN as complex black boxes, is not absolutely valid. In fact, different attempts have arisen to interpret the model's weights or parameters (Masters, 1993), of which the most widely used is the so called time series analysis.

Recurrent Neural Networks

RNNs be situated neural networks with hidden states and loops, permitting information to keep at over time. Now we will first introduce the concept of recurrent units and how their memory works, and finally, explain how they can be used to handle sequence of rain fall time series data. Different types of neural networks such as feed forward networks are created on the idea of learning during training from background and history to produce predictions: RNNs procedure is the idea of hidden state or memory in order to be capable to generate an outcome by updating each neuron into a new computational unit that is able to remember what it has understood previously. This memory is conserved inside the unit, in in a vector, so that when the unit reads an input, it also have procedures the content of the memory, combining the information. By using both (the knowledge from the input and the knowledge from the memory), the unit is now capable of making a prediction and updating the memory itself. RNN units are defined as recurrent units because the type of dependence of the present value on the preceding event is recurrent and can be thought of as manifold duplicates of the similar node, each communicating a recurrent message to a beneficiary.

RNN has an internal hidden state, which can be fed back to the network. In this RNN processes input values and produces output values. The hidden state, allows the information to be forwarded from one node of the network to the next node, organized as sequences of a rain fall time series data. The time series data has information on the current input and previous inputs. The time series data goes through the tanh activation, and the output is the new hidden state. This construction, RNNs can characterize a decent modeling choice to solve a variation of in sequence difficulties, such as forecasting time series, speech recognition, or image processing (Bianchi 2018).There is an additional component denoted as W. It indicates that each unit has three sets of weights, one for the inputs, another for the outputs of the previous time step and the other for the output of the current time step. Those weight values are determined by the training process and can be recognized by applying a popular optimization procedure called gradient descent. Gradients values are used to modernize a neural network's weights: the function characterizes the problem that we want to solve using neural networks and selecting an applicable choice of parameters to update those neural networks weights. In order to compute a gradient descent, then we need to evaluate the loss function and its derivative with respect to the weights. We have seen to be applying the same weights to different items in the input time series data. This means that we are sharing parameters through inputs. If we are not able share parameters across inputs, then an RNN becomes like an ordinary neural network where each input node requires weights of their own RNNs instead can influence their hidden state property that connections one input to the next one and combines this input connection in a serial input. In spite of their great potential, RNNs have a few limitations, as they suffer from short-term memory. We state this difficult is known as the vanishing gradient problem. During back propagation, recurrent neural networks suffer from this problem because the gradient decreases as it back propagates through time. If a gradient value develops extremely small, it is not able to contribute to the network learning procedure so, LSTM network resolves this kind of problem.

Long Short-Term Memory (LSTM)

A type of recurrent neural network called long short-term memory (LSTM) is able to retain values from earlier stages for use in the future. It is important to have an understanding of how a neural network works before diving into LSTM. When modelling time series data, LSTMs can learn long-term dependencies, which is a desirable capacity. In order to prevent the loss of crucial data, LSTMs help to save the error that can be back propagated across time and layers: Gates and cell state are internal components of LSTMs that can control the information flow (Zhang et al. 2019). The cell decides what information to store, how much to store, and when to release it. They discover the appropriate times to permit information to enter, exit, or be deleted by an iterative process of guessing, back propagating error, and gradient descent weight adjustments. It's crucial to remember that LSTM memory cells use addition and multiplication to change input and transfer information. The magic of LSTMs is really the addition step since it keeps the error constant even when it needs to be back propagated at depth. While the forget gate still relies on multiplication, they add the two rather than multiplying the current state of the next cell with new input. We have the input gate to change the cell state. The sigmoid function is used to generate values between 0 and 1 given the information from the prior hidden state and the current input. Values closer to 0 indicate forgetting, while values closer to 1 indicate remembering. Even though their primary function is to convey and maintain data from extreme events to a final output, LSTMs also feature a forget gate. The forget gate makes decisions regarding what information should be transferred and what information should be remembered. The output gate, which determines the subsequent hidden state, comes last. First, a sigmoid function is used to process the current input and the prior hidden state. The tanh function is then applied to the newly altered cell state. The information that the hidden state should retain is determined by multiplying the tanh output by the sigmoid output (Zhang et al. 2019). The LSTM networks, which frequently provide comparable performance and are computed much more quickly.

Evaluation metrics for improving model performance

Four cutting-edge forecasting techniques have been employed in this research as evaluation measures to assess forecasting accuracy. They are employed because the issue is thought to be one of time series forecasting. So, this model aims to predict the amount of rain that will fall over time.

First, MAPE, it is the average absolute percentage error between actual (x_t) and predicted (x_t ')

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{xt-xt'}{xt}\right|$$

Second, MAE calculated the absolute error between actual (x_t) and predicted (x_t ')

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n}|xt - xt'|$$

Third, RMSE is given by

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(\frac{xt'-xt}{n}\right)^2}$$

Finally, R^2 is then used to show how well the dataset fits the data. Additionally, it includes the fluctuation in predicted response variables and actual response variables.

$$R^2 = \frac{\text{sum of square due to residual}}{\text{sum of square due to total}}$$

Training and testing dataset

This study attempts to estimate India's rainfall using monthly climate training data from 1901 to 1985 and testing data from the last 29 years (1986-2015). The chosen models were utilized to predict rainfall over a five-year period (2016-2021). Here, we employed a 75% training and 25% testing split for training and testing. The distribution of monthly rain fall data during the forecasting, testing, and training phases.

Discussion

The review of 115 years' worth of monsoon rainfall data indicates that there is no long-term trend or change in the country's average monsoon rainfall. The overall rainfall in India has not changed, although there have been major changes in the yearly rainfall in several meteorological sub-divisions. There are decreasing trends in rainfall over Kerala, East Madhya Pradesh, Jharkhand,

Arunachal Pradesh, Nagaland, Manipur, Mizoram, and Tripura (NMMT). However, there is a trend toward more rain in coastal Karnataka, Maharashtra, and Jammu & Kashmir. Over India, there is a general trend toward an increase in the frequency of extreme rainfall (heavy rainfall events), particularly over the countries core regions during the southwest monsoon season (June to September). No indication of global warming has been seen in the observed variations in yearly or rainy season across India. However, there is mounting evidence that links global warming to an increase in the frequency of intense rainfall. According to the Intergovernmental Panel on Climate Change's (IPCC) assessment of climate change, India may see more intense rainfall in the future as a result of rising global temperatures. However, there are no additional long-term variations or trends in India's rainfall that may be linked to global warming. It is discovered that the Indian Monsoon is a reliable system. It is particularly challenging for a statistical model to predict rainfall data when there are more variances in the average rainfall in the data. The LSTM neural network is utilized to construct several features that assist for predicting rainfall data that exhibits greater seasonal fluctuation.

The models were calculated with the test data through four forecasting accuracy measures: mean absolute error (MAE), root mean squared error (RMSE), mean absolute percentage error (MAPE) and Multiple coefficient of determination (R^2).The RMSE is the performance measure normally used to estimate LSTM network parameters, mainly thinks to the simple computation involved and its facility for deriving the learning rule (Masters, 1993). Nevertheless, this measure is very sensitive to the presence of large-sized errors and its value is difficult to interpret because it depends on the scale of the measurement of the variable involved. The MAPE is one of the most widely used accuracy measures (Carbone & Armstrong, 1982) because it is easy to interpret – it is interpreted in terms of percentage of error– does not depend on the scale of the measurement of the variable, is not affected by outliers, and is reliable and valid (Law & Au, 1999; Pattie & Snyder, 1996). The comparison between the LSTM model network and the ARIMA model then LSTM shows that the network model produces clearly lower MAE, MAPE, RMSE and higher R^2 values as compared to the classic ARIMA model.

Conclusion and future work

This article presents a procedure known as time series analysis, which allows the most relevant auto regressive lagged terms for times series forecasting to be identified directly and effectively when applied to an LSTM network. It is worth mentioning that ANN have been considered to be devoid of both theoretical baggage and of a systematic procedure for building models, unlike traditional approaches such as the Box-Jenkins methodology (Box & Jenkins, 1976). The most critical aspect in applying ANN is the study of the effect or importance of the input variables on an LSTM network, since the value of the parameters obtained by the network does not have a practical interpretation, unlike a traditional linear statistical model. The time series analysis used in this study attempts to overcome this drawback by allowing the set of lagged auto regressive terms with the greatest forecasting power to be selected numerically. The outcome of our research shows that applying time series analysis allows for a more parsimonious network model that better fits the data, in comparison with the network model derived from the traditional procedure, and that the neural network derived from the time series analysis also outperforms the equivalent ARIMA model. These results allow us to conclude that applying a time series analysis to LSTM networks is an effective procedure for determining the most relevant forecasting variables. As for our studies potential benefits to create a rainfall forecasting model for application in human-based activities, agriculture, transportation, and nearly every aspect of daily life utilizing LSTM neural networks. For certain agricultural industries, being able to predict how much rain will fall is useful. It improves management and decision-making of farming operations, such as planting, irrigation, fertilization, and harvest.

For future research, an alert and decision support system can be developed which can be helpful in flood or draught situation, incorporates numerous parameters for improved predictions may be created, the models can be created for the entire country using monthly, monsoon, and yearly data from the entire nation. And also may focus on the relationship, influence, and impact of climate change and global warming on rainfall, an online learning technique for a neural networks may be utilized to increase the effectiveness of the proposed system.

References

1. Altay, E., & Satman, M.H. (2005). Stock market forecasting: Artificial neural network and linear regression comparison in an emerging market. Journal of Financial Management and Analysis, 18, 18–33.
2. Aznarte, J.L., Benítez, J.M., Lugilde, D.N., Linares, C., Guardia, C., & Sanchez, F.A. (2007). Forecasting airborne pollen concentration time series with neural and neuro-fuzzy models. Expert Systems with Applications, 32, 1218–1225.
3. Bishop, C.M. (1995). Neural networks for pattern recognition. Oxford, UK: Oxford University Press.
4. Box, G.E.P., & Jenkins, G.M. (1976). Time series analysis*:* Forecasting and control. San Francisco: Holden Day.
5. Carbone, R., & Armstrong, J.S. (1982). Evaluation of extrapolative forecasting methods: Results of academicians and practitioners. Journal of Forecasting, 1, 215–217.
6. Castellanos, J., Pazos, A., Ríos, J., & Zafra, J.L. (1994). Sensitivity analysis on neural networks for meteorological variable forecasting. In J. Vlontzos, J.N. Hwang, & E. Wilson (Eds.), Proceedings of IEEEWorkshop on Neural Networks for Signal Processing (pp. 587–595). New York: IEEE.
7. Funahashi, K. (1989). On the approximate realization of continuous mappings by neural networks. Neural Networks, 2, 183–192.
8. Hansen, J.V., McDonald, J.B. &Nelson, R.D. (1999). Time series prediction with genetic-algorithm designed neural networks: An empirical comparison with modern statistical models. Computational Intelligence, 15, 171–184.
9. Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. Neural Networks, 2, 359–366.
10. Incerti, G., Feoli, E., Salvati, L., Brunetti, A., & Giovacchini, A. (2007). Analysis of bioclimatic time series and their neural network-based classification to characterize drought risk patterns in South Italy. International Journal of Biometeorology, 51, 253–263.

11. Kaastra, I., & Boyd, M. (1996). Designing a neural network for forecasting financial and economic time series. Neurocomputing, 10, 215–236.
12. Law, R., & Au, N. (1999). A neural network model to forecast Japanese demand for travel to Hong Kong. Tourism Management, 20, 89–97.
13. Masters, T. (1993). Practical neural networks recipes in C++. London: Academic Press.
14. Masters, T. (1995). Advanced algorithms for neural networks: A C++ sourcebook. New York: Wiley.
15. Palmer, A., Montaño, J.J., & Sesé, A. (2006). Designing an artificial neural network for forecasting tourism time series. Tourism Management, 27, 781–790.
16. Pao, H.T. (2006). Modeling and forecasting the energy consumption in Taiwan using artificial neural networks. The Journal of American Academy of Business, 8, 113–119.
17. Pattie, D.C.,&Snyder, J. (1996).Using a neural network to forecast visitor behavior. Annals of Tourism Research, 23, 151–164.
18. Rzempoluck, E.J. (1998). Neural network data analysis using Simulnet. New York: Springer-Verlag. Schwarz, G. (1978). Estimating the dimension of a model. The Annals of Statistics, 6, 461–464.
19. Sirakaya, E., Delen, D., & Choi, H.S. (2005). Forecasting gaming referenda. Annals of Tourism Research, 32, 127–149.
20. West, P.M., Brockett, P.L., & Golden, L.L. (1997). A comparative analysis of neural networks and statistical methods for predicting consumer choice. Marketing Science, 16, 370–391.
21. Yuval, A. (2000). Neural network training for prediction of climatological time series, regularized by minimization of the generalized cross-validation function. Monthly Weather Review, 128, 1456–1473.
22. Zou, H.F., Xia, G.P., Yang, F.T., & Wang, H.Y. (2007). An investigation and comparison of artificial neural network and time series models for Chinese food grain price. Neurocomputing, 70, 2913–2923.