Time Series Prediction of Online Retail Sales Volume Using LSTM

Xuan Li Yuan Luo Jionglong Su Fei Ma
Mathematical Sciences, Xi'an Jiaotong-Liverpool University, Suzhou, China
Telephone: (86) 512 8816 1633, Email: fei.ma@xjtlu.edu.cn

***Abstract*—As the online retail industry are prevailing nowadays, it is indispensable that real-time and accurate sales prediction are required. In this paper, three models based on conventional Long Short-term Memory Networks obtained: 1) Optimized LSTM using grid search for the perfect hyper-parameters; 2) Hybrid model integrating LSTM with AR model; 3) Hybrid model integrating LSTM with standard variation. All three models achieve the results that over 30% the prediction values are varied within 10% the actual data. In addition, the traditional ARIMA model is introduced to prove the superiority of deep learning. However, it is difficult to tell the perfect model for the whole 80 products, as each model is suitable for some specific types of items.**

I. Introduction

Time series prediction is an effective regression method that exploit the historical data to make a prediction on the future. Several statistic models have been purposed to deal with specific time series with different property [1]. For instance, the Box-Jenkins regression models such as Autoregressive (AR) model and Autoregressive Integrated Moving Average (ARIMA) model focus on the stationary data [2] while autoregressive conditional heteroskedasticity (ARCH) model is normally utilized in exhibition of time-varying volatility [3]. However, in the real world, the large scales of sales datasets are often complex and non-linear corrupted by white noises. Thus, neural networks are introduced.

Since Lapedes and Farber [4] first applied neural network in forecasting in 1987, neural network has become very promising tool. As an information processing system, artificial neural network (ANN) is a complex network of interconnected neurons with the capability of random function approximation through self-learning and self-adapting [5]. Given a dataset, ANN extracts the statistical feature of data, then obtains the functional relationship between variables and observations, generalizes a data model that describes the characteristics of the sample, and then classifies, predict or evaluate the newly given data with the data model learned. A deep neural network (DNN) is an ANN with multiple layers between input and output layers [6]. It can model complex non-linear relationships between variables, and typically, it is a feedforward network which transfers data from the input straight down to the output in one direction.

The Recurrent Neural Network (RNN), is a class of DNNs which can elicit the dynamic temporal behavior of data [7]. The essential feature of this network is that there are both internal feedback and feedforward connections between processing units. Compared to the standard feed-forward neural network which adopts a unidirectional multilayer structure, the RNN has stronger dynamic behavior and computing power [8]. In theory, the RNNs are capable to memorize the previous information to combine with the present, for example, the first data might influence the performance of second data and so on. However, the time interval between the relevant information could be very large, and so RNN might not be able to learn the link between it. This is called the long-term dependency problem [9].

Therefore, the Long Short-Term Memory (LSTM) networks are designed to avoid the long-term dependency problem and make predictions on time series which contains delays between information. Its architecture allows it to remember the long-term behavior of time series. Since proposed]by Sepp Hochreiter and Jürgen Schmidhube in 1997 [**?**], it has raised common interest in the contemporary research fields. The major success acquired by LSTM in the speech applications has proved its potential in recognition. In 2013, LSTM has achieved a record 17.7% error rate on the classic TIMIT natural speech dataset, which to the author's knowledge is the best score recorded [10]. Moreover, abundant literature has demonstrated its feasibility and accuracy in successfully forecasting financial data, such as the stock price [11], S&P 500 index [12] and Bitcoin price [13].

Based on real-time and accurate sales prediction, manufacturers can schedule the production line to avoid overstocking, and manage the inventory effectively. From the perspective of firms, the benefits of sales forecast have been seen in business planning and operation. Researchers have studied sale prediction in the last decades and developed practical quantitative methods [14] [15].

In this paper, we apply LSTM to forecast the sales volume of online retailers such as JD and Amazon. Conventional ARIMA model are processed as the reference to neuron networks. Moreover, In order to improve the efficiency of LSTM, two innovative adaptions are made: a) A hybrid model integrating LSTM with AR model

that emphasizes the correlation among data in each time-step. The parameters of AR(p) model for each item is added as features into the LSTM network for training.

b) Hybrid model integrating LSTM with Standard Deviation that underlines the volatility of data. SDs of each n time-steps form a time series to indicate the fluctuation of dataset, and along with the original sequence, both are inputted in the LSTM network.

II. Methodology

A. Pre-process of data

In this project, a local online retailer with the sales record of 80 products in a duration of 400 days are acquired as the time series. The first 393 days are used to train the model for making prediction in the last 7 days.

B. Autoregressive model

Autoregressive model (AR) is a statistical method of processing stationary time series. Since the time series may not be stationary, the Augmented Dickey-Fuller (ADF) test is introduced [16]. The test is aimed at examining the null hypothesis that there exists a unit root in a time series [17]. Utilizing ADF test to identify the nonstationary series, we carry out d times difference to remove the nonstationary.

Past data X, i.e. from X_1 to $X_t - 1$, are utilized to predict X_t, under the assumption that they form a linear relationship [18]. The notation of an autoregressive model of order p is AR(p) is defined as:

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t \tag{1}$$

where X_t is the current state of time series, c represents constant term, φ_i denotes the auto-regression coefficient of each state, ε_t is white noise (stochastic error which is normally distributed with 0 as means and σ as standard variation).

The appropriate AR(p) model for each time series is selected according to the Akaike information criterion (AIC), a standard to measure the goodness of fit by statistic models [19]. The interaction with the LSTM networks will be explained more precisely in Section II-E1.

C. Autoregressive Integrated Moving Average Model

Autoregressive Integrated Moving Average (ARIMA) Model is a fundamental statistical model in prediction of time series [2]. Indeed, the general ARIMA(p,d,q) model is the combination of AR(p), and MA(q) under d times difference. In this notion p is the lag number of the time series data used in the prediction, while q represents the lag number of the prediction error. The expansion equation of ARIMA(p, d, q) model following:

$$\hat{y}_t = \mu + \phi_1 * y_{t-1} + \cdots + \phi_p * y_{t-p} + \theta_1 * e_{t-1} + \cdots + \theta_q * e_{t-q} \tag{2}$$

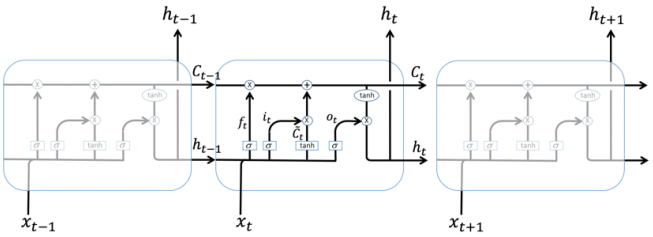

Fig. 1. The structure of LSTM networks

where y denotes value of each time steps, e is the white noise, μ represents the constant value, and ϕ is the coefficient of AR(p) model while θ is the coefficients of MA(q) model.

In this paper, the ARIMA model is the benchmark on which three models are assessed in Section II-D & II-E.

D. Long short-term memory network

Similar to the conventional RNN, the LSTM has the structure of a chain of repeating modules of neural networks. Instead of a single neural network layer in the repeating modules as standard RNN, LSTM has four intersected modules.

In Figure 1, each line carries information in the form of vectors, emerges as the output of one node to become the input into another node. The rectangle boxes represent learned neural network layers, while the circles are pointwise operations such as vector addition or vector multiplication. The merging of lines denotes concatenation and the divergence means that its content is being copied for different purpose.

The horizontal line running through the top of the diagram names cell states, which is the key in LSTM. Information is removed or added to the cell state and is carefully regulated by structures called gates. Gates constitute an approach to select information randomly and determine the allocation of information. The LSTM consists of three gates to process and control the cell states:

1) Forget gate layer f_t: Forget gate layer is a sigmoid layer deciding what information are going to throw away from the cell state.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3}$$

It imports h_{t-1} and x_t and exports a number between 0 and 1 for each component in the cell state $C_t - 1$. A '1' represents 'completely keep this', while a '0' represents 'completely get rid of this'.

2) Input gate layer & Tanh layer: These two sigmoid layers are worked together to decide what new information is allowed to store in the cell state.

$$\begin{aligned} i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i) \\ \widetilde{C}_t &= tanh(W_C[h_{t-1}, x_t] + b_C) \end{aligned} \tag{4}$$

Input gate layer determines what information will be updated, then a tanh layer creates a vector of new candidate values $\widetilde{C}_t$, which could be added to the state. Next, these two are combined to create an update to the state:

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{5}$$

3) Output gate layer: The output layer is a sigmoid layer which determines what parts of the cell state are going to output and then processes it using tanh function in order to transform the values between -1 and 1. In the end, it is multiplied by the output of the sigmoid gate.

$$\begin{aligned} o_t &= \sigma(W_o[h_t - 1, x_t] + b_o) \\ h_t &= o_t * tanh(C_t) \end{aligned} \tag{6}$$

The grid search [20] is used to find the hyperparameters of best prediction performance. It is an exhaustive searching mechanism through a manually specified subset of the hyperparameter space of a learning algorithm. The grid research algorithm must be guided by some performance metric, typically measured by cross-validation on the training set [20]. Thus, in the perspective of each different product, the hyperparameters such as number of epochs, neurons, and lags in the LSTM networks are adopted automatically to minimize the loss function and reach the optimal predictions. Adam [21] is used as optimizer and mean square error is used as the loss function.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2 \tag{7}$$

If $\widehat{Y}$ is a vector of n predictions generated from a sample of n data points on all variables and Y is the vector of observed values of the variables being predicted, then the within-sample MSE of the predictor is computed as above.

E. Ourmodels

In this section, two hybrid models are introduced to reinforce the efficiency of single LSTM network in predicting the future sales.

1) Hybrid model integrating LSTM with AR model (LSTM-AR model): Instead of embedding the ARIMA model into the LSTM networks, the AR model, a linear expression of the previous observations is implemented. The MA model is, from another perspective, discribing the relationship between prediction values and white noises, thus in this study, it is unable to be inserted into the networks.

As we input the training set, the parameters of AR(p) will be modified to achieve the most accuracy fitting and thus form coefficient matrices for every time steps. These matrices of coefficients will be inputted as the features of the original data in the training process of LSTM networks. The coefficient matrices can be regarded as the time series of parameters and will influence the weighting of LSTM networks because the larger amount of input in AR(p) model, the more precise of the coefficients, thus the weight of the latter time steps will increase.

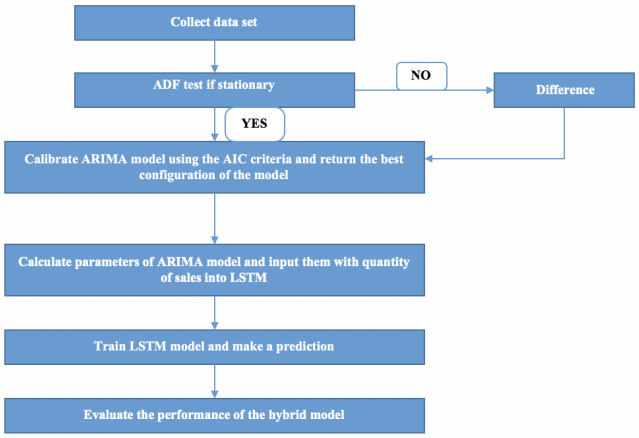

Fig. 2. The flow chart of LSTM-AR model

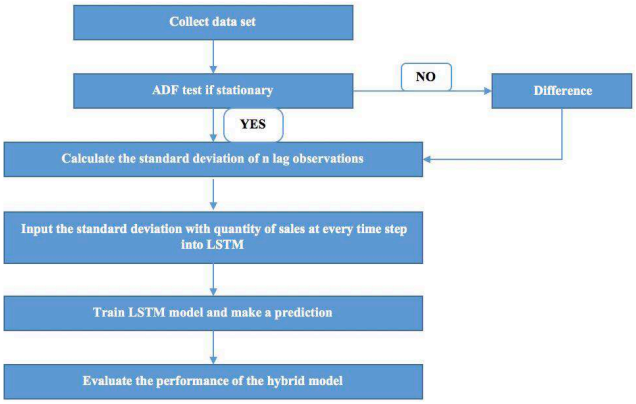

Fig. 3. The flow chart of LSTM-SD model

2) Hybrid model integrating LSTM with SD values (LSTM-SD model): In statistics, standard deviation (SD) is a measure to quantity the statistical dispersion of a set of data. It is formulated as below:

$$s = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N-1}} \tag{8}$$

where x_i is a sequence of numbers, N represents its amount and $\bar{x}$ denotes the average of x_i.

The standard deviation of the previous n time steps is calculated and inputted as the feature of the n^{th} time steps. The addition of the standard variation into the LSTM networks, will reduce the violent fluctuation in the prediction process.

III. Results& Analysis

A. Performance Metrix

We selected 4 metrics to assess the predictive efficiencies of the models. Acc $\pm 10\%$ is to calculate the percentage of 'accurate' predict values which errors are within 0.1 times actual values. Root mean square error (RMSE) and mean square error (MSE) are frequently used to calculate the differences between actual and predicted values, thus to evaluate the performance of models. Similar to the previous ones, mean absolute error (MAE) specifically emphasizes on the real situations of predict errors.

TABLE I
Performance Metrix

Name	Definition
RMSE	$\sqrt{\frac{\sum(a_i-p_i)^2}{N}}$
MSE	$\frac{\sum(a_i-p_i)^2}{N}$
MAE	$\frac{\sum\|a_i-p_i\|^2}{N}$
Acc $\pm 10\%$	$\frac{1}{N}\sum I(\mid a_i - p_i \mid) <= 0.1a_i$

p_i represents the predicted value of ith item, while a_i denotes the actual value of ith item.

B. Results

In Table I,

TABLE II
Results of four models

Models	MSE	RMSE	MAE	Acc$\pm$1*
ARIMA	126.4890781	9.324509094	8.103399714	20%
Optimized-LSTM	133.747815	9.17877288	7.21345103	31.07%
LSTM-AR	245.965625	12.3545423	10.9139463	31.96%
LSTM-SD	260.272437	11.126711	9.66259541	32.5%

References

[1] R. H. Shumway, *Applied statistical time series analysis*. Prentice-Hall, 1990.

[2] R. J. Hyndman and G. Athanasopoulos, "Forecasting: Principles and practice," 2018.

[3] R. F. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation," *Econometrica: Journal of the Econometric Society*, pp. 987–1007, 1982.

[4] A. Lapedes and R. Farber, "Nonlinear signal processing using neural networks: Prediction and system modelling," Tech. Rep., 1987.

[5] M. Van Gerven and S. Bohte, *Artificial neural networks as models of neural information processing*. Frontiers Media SA, 2018.

[6] J. Schmidhuber, "Deep learning in neural networks: An overview." *Neural Netw*, vol. 61, pp. 85–117, 2015.

[7] M. Miljanovic, "Comparative analysis of recurrent and finite impulse response neural networks in time series prediction," *Indian Journal of Computer Science and Engineering*, pp. 180–191, 2012.

[8] R. Rojas, *Neural Networks - A Systematic Introduction*. Springer-Verlag New York, Inc., 1996.

[9] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019*, 2015.

[10] A. Graves, A. R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6645–6649.

[11] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *Plos One*, vol. 12, no. 7, p. e0180944, 2017.

[12] E. Hajizadeh, A. Seifi, M. H. F. Zarandi, and I. B. Turksen, "A hybrid modeling approach for forecasting the volatility of s&p 500 index return," *Expert Systems with Applications*, vol. 39, no. 1, pp. 431–436, 2012.

[13] E. Ş. Karakoyun and A. O. Çıbıkdiken, "Comparison of arima time series model and lstm deep learning algorithm for bitcoin price forecasting," *Proceedings of MAC 2018 in Prague*, p. 171, 2018.

[14] J. N. Mosel, "Prediction of department store sales performance from personal data." *Journal of Applied Psychology*, vol. 36, no. 1, pp. 8–10, 1952.

[15] M. Giering, "Retail sales prediction and item recommendations using customer demographics at store level," *Acm Sigkdd Explorations Newsletter*, vol. 10, no. 2, pp. 84–89, 2008.

[16] G. Elliott, T. J. Rothenberg, and J. H. Stock, "Efficient tests for an autoregressive unit root," *Econometrica*, vol. 64, no. 4, pp. 813–836, 1996.

[17] W. A. Fuller, *Introduction to statistical time series*. John Wiley & Sons, 2009, vol. 428.

[18] T. C. Mills, *Time series techniques for economists*. Cambridge University Press, 1990.

[19] H. Akaike, "A new look at the statistical model identification," *Automatic Control IEEE Transactions on*, vol. 19, no. 6, pp. 716–723, 1974.

[20] C. W. Hsu, "A practical guide to support vector classification," vol. 67, no. 5, 2010.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Computer Science*, 2014.