Future Sales Prediction using Deep learning with long short-term memory networks

Abstract

Future Sales Prediction is a critical component of modern market. A reliable revenue forecast will assist a company in preserving capital on unnecessary product, better preparing for the future, and increasing profit. Future Sales Prediction is associated with predicting the potential sales of stores such as supermarkets and retail outlets. It enables businesses to distribute capital more effectively, forecast realistic sales income, and plan a more robust strategy for the store's potential growth. Traditional forecasting systems cannot compete with big data and revenue forecasting precision. Different data processing strategies can be used to solve these problems. This paper focuses on the challenges of product sales predictive analytics based on historical sell data and time-series analysis. Long Short-Term Memory is used to forecast future sales (LSTM). Proposed model is tested on a real-time dataset obtained from Kaggle (Predict Future Sales Competition).

Introduction

Firms such as manufacturers, distributors, retailers, and so on are always looking for more accurate forecasts in order to reduce uncertainty in decision-making. Accurate demand forecasting, particularly in retail, leads to informed purchasing, inventory management, scheduling, capacity management, assortment planning, and so on. Time-series methods, which attempt to identify trend and cyclicity in the series, are the most commonly used methods for demand forecasting.

(RNN) that is able to effectively capture long-term dependencies in data. RNNs are a type of neural network that are capable of processing sequential data, such as time series, natural language, and speech. However, traditional RNNs have difficulty retaining information about long-term dependencies because they tend to "forget" earlier information as the network processes new data.

LSTMs solve this problem by introducing "memory cells" that can store information for an extended period of time and "gates" that can selectively allow information to be stored or forgotten. In addition, LSTMs are able to handle data with missing values, which is common in real-world time series data. This is because the memory cells and gates in an LSTM model can selectively store and retain information, rather than simply overwriting it as traditional RNNs do.

This enables LSTMs to effectively process and make predictions based on long sequences of data, making them particularly useful in tasks such as in finance, weather forecasting, demand forecasting, language translation, language modeling, and sentiment analysis.

There have been several recent updates and advances in the use of long short-term memory (LSTM) networks for forecasting tasks. Here are a few examples:

- Hybrid models: One approach that has been shown to be effective is the use of hybrid models that combine LSTMs with other types of models, such as seasonal decomposition or autoregressive integrated moving average (ARIMA) models. These hybrid models can capture the long-term dependencies learned by the LSTM and the short-term patterns captured by the other model, leading to improved performance.
- Attention mechanisms: Attention mechanisms have been widely used in natural language processing tasks and have recently been applied to time series forecasting as well. Attention mechanisms allow the model to weight different input time steps differently, allowing it to focus on the most relevant information for making a prediction.
- Multivariate LSTMs: Many real-world forecasting tasks involve multiple time series, such as sales data for multiple products or multiple locations. Multivariate LSTMs can handle multiple input time series and have been shown to outperform traditional univariate LSTM models in these situations.
- Transfer learning: Pre-trained LSTM models that have been trained on large amounts of data in a related domain can be fine-tuned for a specific forecasting task, allowing for the transfer of learned features to the new task. This can be especially useful when data is limited or hard to obtain.

These are just a few examples of the many recent developments in the use of LSTMs for forecasting. There is ongoing research in this area and it is likely that new techniques and approaches will continue to emerge.

Proposed Method

The LSTM network

The selection of the LSTM network for forecasting in retail is based on several reasons. Some of them are as follows. LSTM networks recently have shown promising results in time-series forecasting tasks (Fischer and Krauss 2018). LSTM networks are capable of working well on linear and non-linear time series (Chollet, 2017).

The Architecture of an LSTM network memory cell

LSTM networks belong to the class of recurrent neural networks (RNNs). RNNs have the property of information persistence - i.e., retaining the state variables across time steps (Graves and Schmidhuber 2005), thus making sequential learning over time steps feasible. The architecture of an RNN is presented in Figure 1, where X_t is the input, S_t is the hidden state of the cell and H_t is the output of the RNN cell at time t. RNN can only handle short-term dependencies because it suffers from vanishing gradient problem. The LSTM networks, on the other hand, have the capability to learn long-term dependencies (Hochreiter and Schmidhuber, 1997).

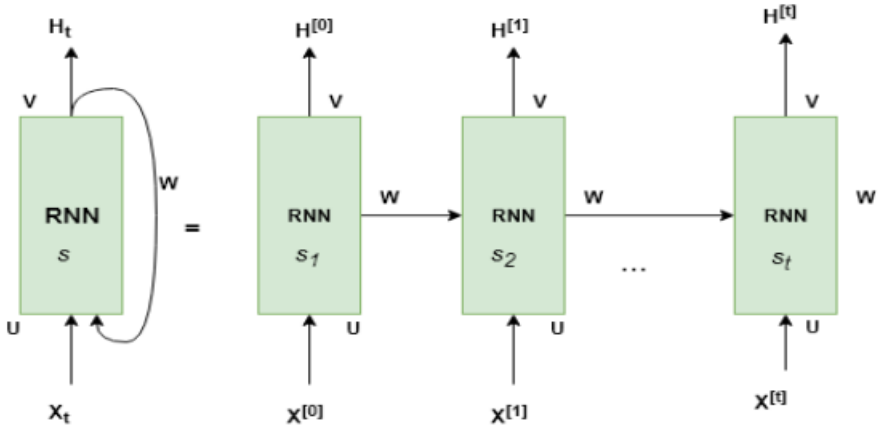


Figure 1: Recurrent Neural Network

The architecture of the LSTM network has three types of layers:

1) An input layer with a number of neurons equal to the number of input variables.

2) Single or multiple hidden layers

3) An output layer with a number of neurons equal to the number of output variables. The hidden layers of LSTM networks consist of a memory cell. LSTM networks are superior to standard RNN due to the presence of this memory cell, which helps to retain information across time steps as this was not possible in earlier neural networks. The structure of the memory cell has three types of gates: 1) a forget gate (f_t), 2) an input gate (i_t), and 3) an output gate (o_t). We can see the complete architecture of the memory cell in Figure 2. In memory cell, at each time step t, the input consists of an element from the input sequence (X_t) and the output of the previous time step (h_t-1). At cell state t: a) the forget gate takes these inputs and decide upon which information will be removed from memory, b) the input gate decides which information shall be added to memory (at cell state t), and c) the output gate decides the output of the memory block.

X_t: the input vector at time step t

h_t : the output vector at time step t

s_t : the vector for cell state t

$\tilde{s}_t$: the vector for a candidate value for input gate

$b_f, b_i, b_{\tilde{s}}, b_o$: bias vectors

$\sigma(.)$: denotes the sigmoid function $\left(f(x) = \frac{1}{1+e^{-x}}\right)$

$W_{f,x}, W_{f,h}, W_{\tilde{s},x}, W_{\tilde{s},h}, W_{i,x}, W_{i,h}, W_{o,x}, W_{o,h}$: weight matrices for input and outputs for the three gates

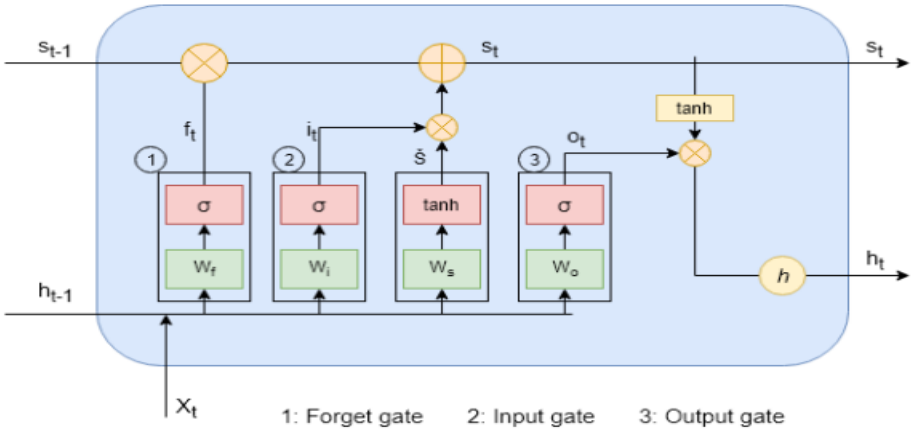
f_t, i_t, o_t : vectors of values obtained after activation of the gates (forget gate, input gate and output gate)

LSTM network processes information through a sequence of four steps:
In the first step, a sigmoid (a non-linear activation function) layer called forget gate layer, which takes X_t and h_{t-1} as inputs and bf as bias, computes

the vector of activation values, f_t, for each of the values in cell state s_{t-1} within a normalized range between 0 (completely get rid-off) to 1 (completely keep).

Then the activation value vector is calculated as follows:

$$f_t = \sigma\,(W_{f,x}\,X_t + W_{f,h}\,h_{t-1} + b_f) \qquad (1)$$



	Product 1								
Week	1	2	3	...	101	102	103	104	...
Units	27	36	28	...	48	44	30	27	...

	Product 1					
Seq. 1	1	2	3	...	101	102
	27	36	28	...	48	44

	Product 1					
Seq. 2	2	3	...	101	102	103
	36	28	...	48	44	30

Figure 2: The LSTM Memory Cell Architecture and the input data to the LSTM

In the second step, it is decided which information will be added to the memory cell state s_t. This step has two parts: first, candidate values $\tilde{s}_t$ are calculated. In the second step, an activation layer called the input gate layer, calculated as follows:

$$\tilde{s}_t = tanh\,(W_{\tilde{s},\,x}\,X_t + W_{\tilde{s},\,h}\,ht{-}1 + b_{\tilde{s}t}) \qquad (2)$$

$$i_t = \sigma\,(W_{i,x}X_t + W_{i,h}h_{t-1} + b_i) \qquad (3)$$

In the third step, we update the cell state using new information. We use the Hadamard product in this step:

$$s_t = f_t\,.\,s_{t-1} + i_t.\,\tilde{s}_t \qquad (4)$$

In the last step, ht , we calculate the output of the memory cell as follows:

$$o_t = \sigma\,(W_{o,x}X_t + W_{o,h}h_{t-1} + b_o) \qquad (5)$$

$$h_t = o_t * \tanh\,(s_t) \qquad (6)$$

As shown in Figure 2, the input variables (demand and lagged variables) are inserted to the LSTM's input gate. LSTM network process the inputs step by step using Equations (1)-(6), and after completion of the process, it generates the final output sequence. The output vector is then the forecast of the network. LSTM networks are trained in multiple iterations known as epochs. During these iterations, bias and weights change to minimize the objective function across the training sets. For our task, the mean absolute error (MAE) is used as loss functions and advanced hyper-parameter optimization through grid search is used to decide the final parameters of the prediction model (Bergstra, Yamins, and Cox 2013).

Outputs

Dataset

We conduct experiments on the dataset, Kaggle Predict Future Sales Datasets, which consisting of sales, shop items and information about the items and shops from January 2013 to October 2015. The training set is the daily historical data from January 2013 to October 2015. The test set is to forecast the sales for products for November 2015. The statistical information of datasets is detailed in Table 1.

Dataset	Unique Values
Shops Dataset	60
Items Dataset	22170
Item Categories Dataset	84
Sales Train Dataset	1034
Test Dataset	214000

Table 1. Information of datasets

Then we started merging the data using joins to form a single dataset and checking for Null values. As shown in figure 3 Heatmap doesn't give us very clear picture, but we can say we don't find much correlation with the target column.

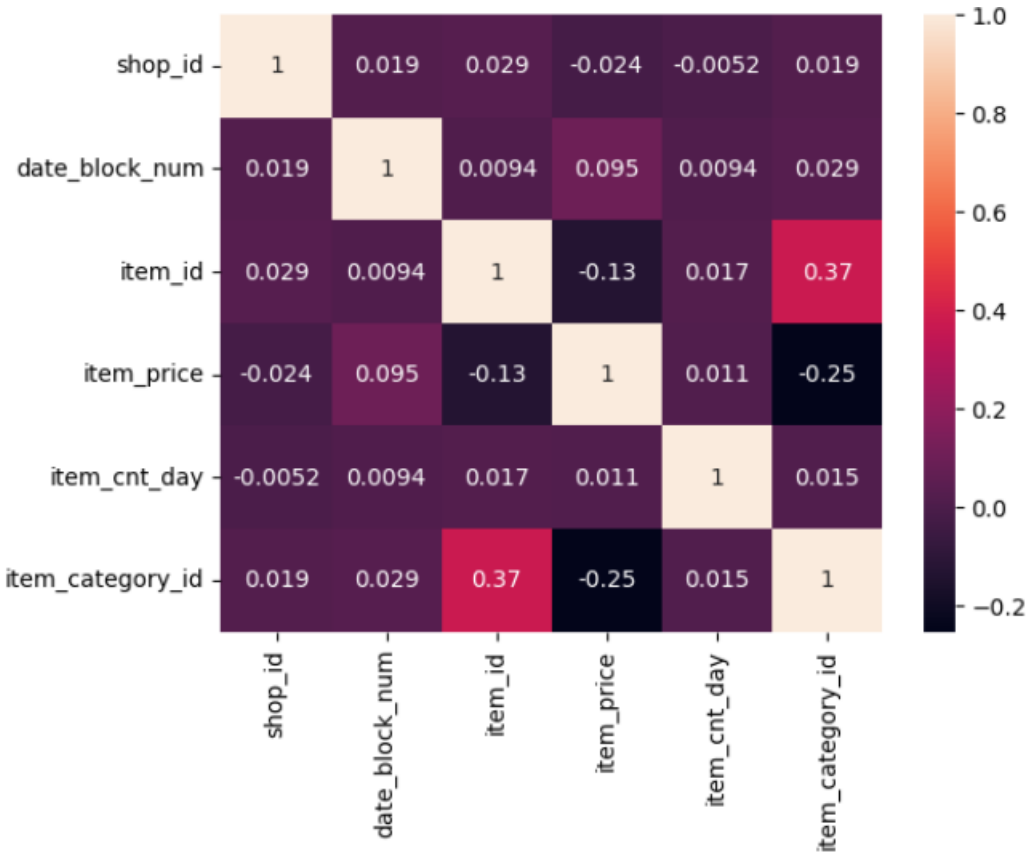


Figure 3. Heatmap

Data Preprocessing

We Started label encoding the categorical variables and using datetime to convert the date into datetime format. Then we created a Dataframe with a date column and converted the date column to datetime. We extracted the month in numeric form to create a sequence of numbers and then we dropped unnecessary columns and re-arrange the dataset. After that we removed the negative values of target variable, checked if we have got any outliers as shown in figure 4 and removed these outliers using (Z-SCORE).

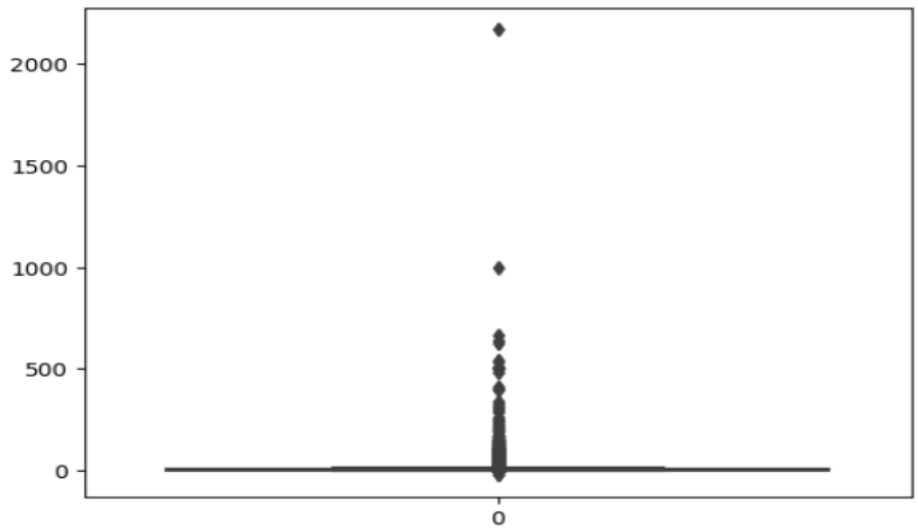


Figure 4. Outliers Graph

Model setup

Keras (Library), Adam (optimizer), Sequential (Model) and Relu (Activation Function) will be used for LSTM model. Lag Feature will be used to calculate the difference between (two dates in months and two dates in days) and calculate the number of months between the two dates. As shown in figure 5 we have plotted a scatter plot (Monthly sales) between Item count per day and Month.

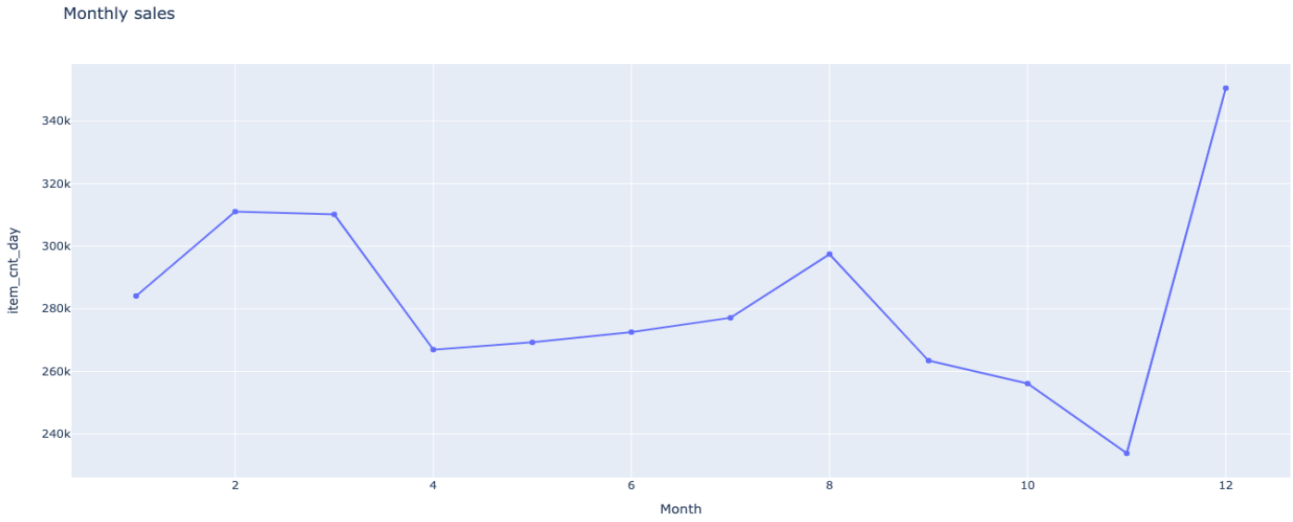


Figure 5. Scatter Plot

The main experimental parameters of forecast model are shown in Table 2.

Parameters	Values
Patch size	256
Learning Rate	0.0003
Epochs	40
Number of LSTM layer	50
Number of Dense layer	1

Table 2. Experimental Parameters

Result

After carrying out the model on the dataset, we will get a RMSE (Root Mean Square Error) equal to 0.778 and a RMAE (Root Mean Absolute Error) equal to 0.716. The RMSE Result is between the True target values that are clipped into [0,20] range.

Conclusion

In this paper Long Short-Term Memory is used to forecast future sales (LSTM). Proposed model is tested on a real-time dataset obtained from Kaggle (Predict Future Sales Competition). Using LSTM we have predicted the sales (Quantity of each Item sold in November 2015) in this paper and also we have calculated the RMSE and RMAE.

References

2020_Deep_learning.pdf (bangor.ac.uk)

https://ieeexplore.ieee.org/document/9587668

https://www.researchgate.net/publication/349111600_A_Sales_Prediction_Method_Based_on_LSTM_with_Hyper-Parameter_Search

https://www.kaggle.com/competitions/competitive-data-science-predict-future-sales/data