2020 International Conference on Computer Science and Management Technology (ICCSMT) | 978-1-7281-8668-9/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICCSMT51754.2020.00088

Automatic Sales Forecasting System Based On LSTM Network

XINJIE LI
Donghua University,
Shanghai, China,
Cynthiaplusss@gmail.com

Jiakai Du
Shanghai Maritime University,
Shanghai, China
15821701571@163.com

Yang Wang
University of Sydney,
Sydney, Australia,
228818651@qq.com,

Yuan Cao
The University of Manchester
Manchester, England
katetsao0171@outlook.com

***Abstract*— Sales forecasting is especially valuable for retail business owners. Based on sales predictions, shops can greatly reduce actual or opportunity costs and generate more revenue. In this paper, we proposed a long short term memory(LSTM) based deep learning method to predict future sales in 28 days according to previous sales records. Our model is trained and evaluated on the Walmart sales dataset which contains sales records for 30491 goods over a course of 1913 days among supplementary information. We employed efficient feature engineering techniques and in experiments, our model outperforms other methods like SVM and Linear Regression, reaching an RMSSE of 0.834.**

Index Terms—Walmart sales forecast, LSTM, Feature engineering

I. INTRODUCTION

Sales forecasting systems use previous sales records of products and other supplementary information to predict future sales. Such systems influence retail owners, whether their business is large or small, on all levels of decision making. This includes budgeting, supply chain planning, stock management, promotion planning, and even shelf placements. Accurate predictions can help business owners maximize the use of budget, storage, shelf space, and staff, bringing in more revenue. While lousy predictions may cause an actual loss on the storage of overstocked goods and opportunity loss on selling less desired products.

The great importance of sales prediction systems in the retail industry has attracted a lot of research attention. Many previous works have been done in this field. But most of them suffer from the lack of sufficient data to test the generalization ability of their models. Also,traditional machine learning models don't have structure specifically for processing time-series data.

A. Related Work

The problem of sales prediction can be approached as a linear regression problem, where the inputs are previous sales records and the system outputs a continuous value as prediction subsequently.

To solve this problem, Volodymyr Kuleshov et al.[1] used the Calibrated Regression method, refining the predictions of a Bayesian Neural Network to predict sales. Bayesian uncertainty estimates often fail to capture the true data distribution[2]. So they calibrated the results to reduce the variance uncertainty that brings to 90% posterior credible interval predictions.

In [3], Glib Kechyn et al. proposed the use of WaveNet, a convolutional neural network(CNN) architecture on sales prediction. N days of sales record along with other background information are extracted from the dataset as training input, and the sales record for the day after is used as model output. Their method achieved second place in Kaggle Corporación Favorita Grocery Sales Forecasting competition[4]. However, CNN's architecture treats all input data with the same convolutional structure, thus is unable to effectively capture the temporal relationship among data from different days.

Oscar Chang et al.[5]'s work employs a pipeline including three shallow deep neural networks(DNN) and an autoencoder. The model is trained in slide window fashion on pharmaceutical products sales data and makes weekly forecasting. Their model is more accurate than previous work, but the overly complex training pipeline means the models need a large amount of data to train, which is expensive to get.

Weng et al.[6]'s work combines LSTM and LightGBM. The machine learning model needs more human intervention in the feature selection process, increasing the human cost of applying such a method. But it also improves the interpretability of the model and allows humans to incorporate domain knowledge into the system, which is more valuable on small datasets.

In this paper, we present an accurate forecasting system based on LSTM along with efficient feature engineering techniques to reduce the computational resource our method takes. The LSTM nodes in our method can more effectively utilize temporal information than traditional machine learning methods and make better predictions.

B. Our Contribution

In this research, we used the M5 Forecasting - Accuracy dataset from Kaggle. It contains sales records for 30490 products over a course of 1913 days. The data is collected from 10 Walmart department stores across 3 states. Details about the products like it's department and categories along with details about promotion activities in this time period are also included. During data preprocessing, we firstly removed products with the lowest 0.5 percent average sales. Then we did feature engineering to generate statistical and aggregative features like average sales for the past month or sales during the last

978-1-7281-8668-9/20/$31.00 ©2020 IEEE
DOI 10.1109/ICCSMT51754.2020.00088

promotion to give our model contextual information over a longer time period. After that, we filled in missing values and one-hot encoded all the categorical data. At last, we downcasted the data types of all features, reducing the memory consumption to a fourth of the original size. Our model is made up of three LSTM layers and a Dense layer. The loss we choose is the mean squared error(MSE) loss. Important model hyperparameters, including learning rate, LSTM time step, epoch number, and else are chosen with the grid search technique. Our model achieves 0.834 RMMSE, outperforming the Linear Regression and SVM methods.

II. FEATURE ENGINEERING

Feature engineering is the first step in all machine learning systems. Most of the time, it includes data sifting, feature generation, encoding, automatic, and manual feature selection. The M5 Forecasting - Accuracy dataset consists of three tables: calendar, product sales, and product price. The calendar table contains detailed information about the 1913 days over which time the sales data is collected. It includes the year, month, day, week of the year, weekday, and most important event name and event type. Promotion events influence the sales of products greatly. Information about promotion events contributes greatly to prediction accuracy. The product sales table contains sales information of the 30490 products over the 1913 days. It also includes details of the product like it's department and category. Besides, details about the store that products are sole in including store_id and the state the store is in are also provided. At last, the product price table contains the prices of the product for each day during the recorded period. A discount may also trigger changes in sales thus this is also valuable information.

The first step of our feature engineering process is removing products with the lowest 0.5 percent average sales. Most of these products are sold less than once a month if not sold at all. We don't want such extreme cases to influence the general performance of our model. Then we generated aggregative or statistical features to help our model get more contextual information over a longer time period. For the product sales table, we generated average sales over the last 7 days as a baseline for future predictions. From our observation, special occasions like promotion or weekend also influence sales greatly. So we generate average sales for each day under past similar occasions. For days with a promotion going on, we take the average sales of the product in 4 nearest promotions of the same category. For normal days we calculate the average sales for the last 4 same weekdays. In the product price table, we consider the variations in product price the principal factor affecting product sales. So we add the change in product price compared with the previous day to each column in the product price table. Finally, we one-hot encoded all the categorical features in the dataset and downcaseted the data types of features to the minimum bits required to store all the information. We can reduce the memory consumption of our training dataset from 452MB to 114MB. For large retailers like Walmart who sell more than 46,000,000 products, this can be a huge save on data storage cost.

III. LSTM MODEL

This section introduces our LSTM based deep learning model along with the grid search hyperparameter selection technique.

LSTM nodes are a kind of deep neural network(DNN) structure that has loops in them. This means they can take in hidden states from the same node during the previous feed-forward computation. When this structure is unrolled, LSTM is not too different from other DNN structures. They also take input from shallower layers of the DNN and aside from that, they take input from themselves. In our case, during the computations done for predicting sales of the previous day. We apply decay parameters on the loop passage. As a result, information with a longer history will gradually fade away in LSTM's hidden states. This means the network can learn how much temporal contextual information to preserve based on the characteristics of the training dataset. For example, sales information from more than five years ago may have virtually no value in predicting sales for tomorrow. This way, our network can utilize the limited hidden state representation ability more efficiently to memorize the more valuable, fresh information.

The backpropagation process of the LSTM model is more complex than normal DNN structure, the gradient flows not only backward in the network structure, from deep to shallow, but also backward in time. Also, the hidden states and various gated units mean the LSTM node has more parameters. As a result, we decide to keep our model small and streamline to avoid overfitting the training dataset with an overly complex model. The simple model also performs well on smaller datasets. After various experiments, we decided to include three LSTM layers with 50, 400, and 400 nodes and one dense layer with 30490 nodes as output. The structure of our model is illustrated in Fig 1.

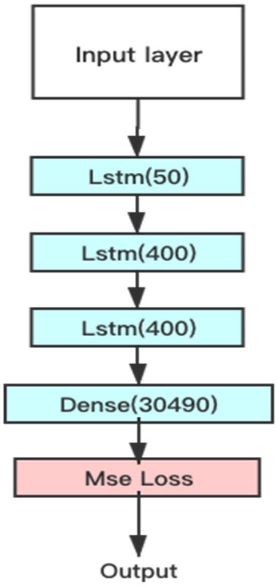


Figure 1. MODEL ARCHITECTURE

For vital model hyperparameters, we employed a grid search technique to search for the most adequate values. Like the learning rate, timestep for LSTM, and training epoch. Grid search builds a high dimensional hyperparameter space where each parameter is treated as an axis. It firstly searches through this space evenly, then focuses on places showing promising results. By searching more and more densely, it finally comes up with an optima hyperparameter selection.

IV. EXPERIMENTS

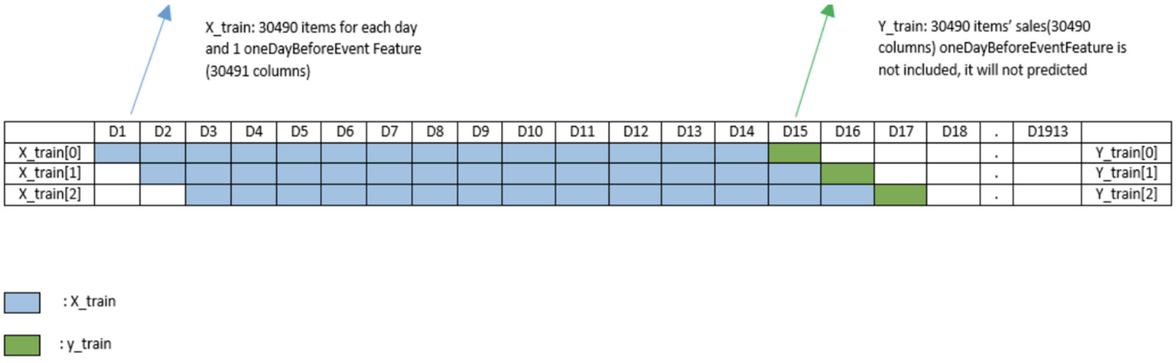
In this section, we describe how we train our model and evaluate it together with Linear Regression and SVM methods.

Our grid search method decides the best timestep for the Walmart dataset to be 14. So we take the sales record of every 14 days and use the 15th day's record as output. This process is shown in Fig 2. The training batch size is set to 44 based on our GPU memory size. We used mean square error loss as the model loss function and trained for 32 epochs.

We also trained a Logistic Regression model and an SVM model on the training set. Predictions are done on the evaluation dataset and submitted to Kaggle to be evaluated. Table 1 shows the results. It is clear that our model outperforms the other two models in RMSSE, reaching the lowest score of 0.834

TABLE I. PERFORMANCE COMPARISON OF MODELS

Models	**RMSSE**
Logistic Regression	1.46
SVM	1.12
LSTM	**0.834**



X_train: 30490 items for each day and 1 oneDayBeforeEvent Feature (30491 columns)

Y_train: 30490 items' sales(30490 columns) oneDayBeforeEventFeature is not included, it will not predicted

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	.	D1913	
X_train[0]	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟩				.		Y_train[0]
X_train[1]		🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟩			.		Y_train[1]
X_train[2]			🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟩		.		Y_train[2]

🟦 : X_train

🟩 : y_train

Figure 2. TRAINING PROCESS. 14 DAYS' SALES RECORD ARE USED TO PREDICT THE 15TH DAY'S SALES

V. CONCLUSIONS

In this paper, we presented a LSTM based deep learning model for the task of product sales predictions. Firstly, we described the effective feature engineering technique that we use. Then our model, the LSTM node, together with the grid search hyperparameter choosing process are described in detail. In the final section, we compared the experiment results of our model against two other methods Linear Regression and SVM. The results show that our model outperforms other methods in RMSSE score. Our work has shown that LSTM's recurrent structure is the most suitable machine learning tool for capturing time series information and it performs the best in product sales prediction tasks.

REFERENCES

[1] Kuleshov, V., Fenner, N., & Ermon, S. (2018). Accurate uncertainties for deep learning using calibrated regression. arXiv preprint arXiv:1807.00263.

[2] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. arXiv preprint arXiv:1612.01474, 2017.Tibor Kiss, Claudia RochJan, Jan Strunk.A logistic regression model of determiner omission in PPs.2010.

[3] Kechyn, G., Yu, L., Zang, Y., & Kechyn, S. (2018). Sales forecasting using WaveNet within the framework of the Kaggle competition. arXiv preprint

[4] arXiv:1803.04037.https://www.kaggle.com/c/favorita-grocery-sales-forecasting

[5] Chang, O., Naranjo, I., Guerron, C., Criollo, D., Guerron, J., & Mosquera, G. (2017). A Deep Learning Algorithm to Forecast Sales of Pharmaceutical Products. no. August.

[6] Weng, T., Liu, W., & Xiao, J. (2019). Supply chain sales forecasting based on lightGBM and LSTM combination model. Industrial Management & Data Systems.

[7] Xiang Wei, Mengzhong Ji, Jun Peng.The application analysis of forecasting housing monthly rent based on Xgboost and LightGBM algorithm.2019.