A Study on the Deep Learning based Prediction of Production Demand by using LSTM under the State of Data Sparsity

Ho-Seok Jung¹ and Sungbum Park¹,*

¹ Graduate School of MoT, Hoseo University, 20, Hooeo-ro 79 beon-gil, Baebang-eup, Asan-si, Chungcheongnam-do, Republic of Korea 31499

*Corresponding author’s e-mail: parksb@hoseo.edu

Abstract. Rapid market condition changes and growing uncertainties require accurate product demand forecasting to reduce the risks of business operations. So far, recurrent neural networks (RNNs) have been introduced as the representative models for forecasting time series data. However, it showed limited performance when learning time pattern occurring over a long period of time. This study performs deep learning-based production forecasting analysis with Long-term short-term memory (LSTM) and dropout algorithm as the alternative for the conventional time series analysis methods and the exponential smoothing method. Our result provides the reference for sales target setting, facility investment and production planning, inventory control, supply chain management, and marketing strategy establishment.

1. Introduction

Accurate product demand forecasting enables companies to strategically implement the key elements of overall corporate management, such as supply chain management (SCM), facility investment planning, and maintenance of appropriate inventory levels [1].

Recently, recurrent neural networks (RNNs) have been introduced as models for forecasting time series data such as written language and speech sound, and product demand. However, conventional RNN model showed limited performance when learning the specific patterns in time series data occurring over a long period of time. To overcome this problem, long short-term memory (LSTM) has been proposed. The concept of memory introduced by LSTM enables patterns recognition in time series data over a long period, which has led to improvement in the RNN performance.

However, prediction using LSTM based deep learning model has hardly been successful so far, either. This is due to the fact that time-series data for event forecasting of interest generally has a sparsity problem. For example, for data occurring once a year, there exist only five samples for five years. In addition, the demand patterns five years ago are highly likely to differ from the present demand patterns. So, the representativeness of the sample also decreases.

Meanwhile, various learning methods have been proposed along with LSTM in order to increase the forecasting ability. Among the algorithms, the most commonly used one is dropout. The dropout rules out some nodes in a deep network with the starting probability (for example, 0.6) during the learning process. Dropped-out nodes are randomly selected each time. The output value for the current connection strength (weight) is calculated with dropped-out nodes. Once completing learning, the output value for new data is calculated by multiplying the output values of all nodes by 1 – previous probability without omitting the nodes. This approach improves the stability and accuracy of learning [2].
In this idea, this study attempts to apply the LSTM with dropout method with the aim of verifying its applicability on the demand forecasting. The result of this study shows how dropout actually works and how corporations utilize the experimental results through demonstration.

2. Theoretical review

2.1. Changes in the methods of time series analysis

Deep neural networks (DNNs) use more intermediate layers than conventional neural networks to improve the capacity to represent data. Currently, representative models of deep learning include convolutional neural network (CNN) models and recurrent neural network models, and recurrent neural networks are known to be suitable for learning and reasoning for time series data.

A recurrent neural network (RNN) is a neural network structure used for modelling sequence data. The basic idea underlying the RNN model is that since each data after the present data in the sequence provides new information, the information is used to update the present state of the model. When a sentence of a text is read, the data of the present state is updated with each new word, and this state is dependent on the previous words as well as the newly introduced word.

RNN models are based on the concept of a chain structure, and there are various types of RNN models depending on how data are maintained and updated. As the word ‘recurrent’ in the term suggests, RNN is composed of a kind of ‘loops.’

Although RNN processes the current input data in connection with the previous input data, it has a limit in remembering the input data that was processed very long time ago. According to Bengio et al. [1994], when learning for long sequences of input data is performed using BPTT (Backpropagation Through Time) in a RNN model, the vanishing gradient problem or the exploding gradient problem may occur. These problems occur because the gradient value vanishes or increases exponentially because of the recurrent weights that are repeatedly multiplied during the process of learning of even the data of the distant past.

The methods to solve the vanishing gradient problem, extended RNN models such as Long Short-Term Memory (LSTM) or gated recurrent units (GRU) have been proposed as fundamental solutions. LSTM was first introduced in 1997 and is now one of the most widely used models in natural language processing. GRU, which was proposed in 2014, is a simplified version of LSTM. Both models were introduced to solve the vanishing gradient problem and are known to handle long sequence data effectively.

2.2. Long Short-Term Memory (LSTM)

Gradient descent, which is a learning method of RNN, uses the change of gradient. One strength of RNN is that it connects the present task to previous information. However, RNN has a limitation in considering long-term dependency of data due to the vanishing gradient problem, which refers to the loss of major information as it goes through several time steps when processing long sequences in the learning process. Therefore, studies on LSTM have been conducted to solve the long-term dependency problem of RNN.
LSTM is composed of four layers with a recurrent structure as shown in Figure 1. The core of LSTM is a continuous cell state coming through the gate. The continuous cell state is also called the conveyor belt. The information coming through the conveyor belt is transmitted without any changes. LSTM can add or delete information through the input gate, forget gate, and output gate. In short, the gates perform the role of selectively transmitting data and thereby remove previous data to continue learning. LSTM performs calculation using gate vectors.

\( f_t \) is the forget gate vector and serves as the weight to remember the previous cell state, and it is the input gate vector and serves as the weight to acquire new information. On the other hand, \( o_t \) is the output gate vector, and performs the role of selecting output candidates. \( \times_t \) is the input vector and, \( h_t \) is the output vector. \( c_t \) is the cell state vector, \( W \) and \( U \) are parameter matrices, and \( b \) is a vector, respectively. \( f_t, i_t, o_t \) are gate vectors. Two types of activation functions are used in LSTM: \( \sigma \) signifies a sigmoid function, and tanh represents the hyperbolic tangent function \([4]\).

### 2.3. Recurrent neural network and dropout

Among various algorithms that enhance the learning capabilities of deep networks, dropout is the algorithm that has received the most attention and is currently commonly used.

In the initial version of dropout proposed for RNN and LSTM, recurrent connections are maintained as they are and dropout is applied only to non-recurrent connections (Figure 2 and Figure 3). This method is to omit some part of connection lines instead of dropping out nodes as in the conventional dropout. As a result, this technique led to improved performance in tasks such as handwriting recognition, but showed poorer performance in phoneme recognition, compared to other normalization techniques such as weight noise injection \([2]\).

Dropout for RNN and LSTM, that is, rnnDrop distinguishes it from existing dropout algorithms for RNNs is to apply dropout not to connections but to nodes themselves. RnnDrop is also different from existing dropout methods for RNNs in that rnnDrop does not generate a new dropout mask for each time frame, but a dropout mask is generated once for the entire sequence and the same mask is applied to all time frames \([2]\).

This method is more natural than conventional dropout methods for RNNs in that one dropout mask is used for one sequence, just as one random dropout mask is used for one image. Moreover, in order to train an RNN model, the recurrent layer is generally unfolded to apply backpropagation (backpropagation through time or BPTT), as shown in Figure 4, and at this time, unlike other dropout methods, rnnDrop allows the same neural network connection structure to be maintained in the unfolded neural network at each time step even after its application.
3. Methodology and results

3.1. Data collection and preprocessing
Corporation H makes available the data of production plans, the previous day’s production performance, and the inventory quantity to its partners through its website. The production plan refers to the planned production quantity that Corporation H plans to produce on the retrieval date and the next day (the company shares the hour-by-hour updates of the production plan, but this study reflected the total daily quantity). The production performance of the previous day represents the amount of materials used for production of Corporation H on the previous day. The inventory quantity is the quantity of materials that Corporation H has in its possession. The production plan, inventory quantity, and the previous day’s production performance distributed by Corporation H on the retrieval date were collected from January to June 2018. In addition, the shipment quantity sent from D company to the customer company, which is the target variable in this study, was collected from January to June. The data was sorted and arranged by product category to be analyzed as shown in Table 1.
### Table 1. Research variables and statistics

| Variable | Category | Description                                | Min | Mean | Max | Kurtosis | Skewness |
|----------|----------|---------------------------------------------|-----|------|-----|----------|----------|
| Month    | numeric  | Month of occurrence                         | 1   | 3.01 | 5   | -1.31    | -0.02    |
| DAY      | numeric  | Date of occurrence                          | 1   | 15.62| 31  | -1.19    | 0.01     |
| ENGINE   | numeric  | Major category                              | 0   | 4.08 | 8   | -1.48    | 0.01     |
| Product  | numeric  | Specific item                               | 0   | 25.50| 51  | -1.20    | 0.00     |
| Demand   | numeric  | Quantity demanded by Corporation H          | 0   | 261.05| 4,792| 11.07    | 3.35     |
| Act      | numeric  | Quantity supplied by D company to Corporation H, considering Demand, BeAct, stock, etc.| 0   | 276.23| 4,536| 9.88     | 3.11     |
| BeAct    | numeric  | Stock of Corporation H Before Act           | -6  | 268.37| 4,356| 9.82     | 3.08     |
| Stock    | numeric  | Stock of Corporation H                      | -558| 151.01| 3,808| 22.30    | 4.16     |
| SDemand  | numeric  | Cumulative Demand                           | 0   | 20,708.65| 362,166| 13.03    | 3.39     |
| SAct     | numeric  | Cumulative Act                              | 0   | 19,610.98| 358,312| 14.71    | 3.49     |
| SBeAct   | numeric  | Sdemand – Sact                              | -132,246| 1,097.67| 72,836| 32.63    | -0.65    |

Assuming that D company is notified by Corporation H today that the next day’s Demand is 170, Act represents D company’s performance of delivering parts to Corporation H’s warehouse in consideration of BeAct and stock as well as the Demand. With respect to Act, BeAct and stock which occurred because D company did not produce products according to Demand even though Corporation H distributed the information of Demand, were also taken into consideration.

#### 3.2. Data analysis

In general, Corporation H has independent factories for each type of engine. A production line of each factory manufactures several products of the same engine product category. Since products in the same product category are expected to have a strong impact on each other in terms of demand, this research set product category as the unit of analysis. Accordingly, during data pre-processing, data was divided according to the product category, and demand forecasting was performed separately for each product category.

The input layer was composed of Demand, BeAct, stock, SDemand, SAct, SBeAct, and Act, and the hidden layer was set as a single layer considering that events were sparse and there was small data available for use. DNN models were trained by randomly sampling 80% of the response-to-demand data, and the experiments were repeated 52 times to derive verification statistics by applying trained models to the tests of the remaining 20% of data [5].
Table 2. Hyper parameters for LSTM model

| Option               | Setting                      |
|----------------------|------------------------------|
| Input Dimension      | 6                            |
| Output Dimension     | 1                            |
| Sequence Length      | 151                          |
| Hidden layer         | 1                            |
| Loss function        | Mean squared errors (MSE)    |
| Optimizer            | AdamOptimizer                |
| Activation function  | Rectified linear unit (ReLU) |
| Dropout ratio        | 0.0                          |

Figure 5 shows the flowchart of the LSTM model for time series forecasting.

The Google TensorFlow library was used and the DNN model was set through the optimization process. The LSTM model was trained by randomly sampling 80% of the 153 matchup datasets, and the trained model was applied to the remaining 20% of the matchup data to calculate the amount of evapotranspiration, and the obtained value was compared with the ground observation data. These steps were considered to comprise a single experimental procedure, and this experiment was repeated 200 times to derive verification statistics [5].

As shown in Table 2, Adam (Adaptive Moment Estimation) was used as the optimizer to minimize the loss function and improve the learning speed. Also, ReLU (Rectified Linear Unit) was used as the activation function that solves the vanishing gradient problem of the loss function that can occur in the backpropagation process (Agarap, 2018). In this study, the DNN model was built by optimizing the weights and bias set using the backpropagation algorithm, and then, dropout was performed again to improve optimization [2] (Choi & Min, 2015). Table 3 shows the result. In the situation where data sparsity exists, it can be seen that the prediction accuracy of the model improves as the dropout is not applied to the LSTM model.
Table 3. LSTM analysis result according to various dropout ratio

| Prod No | Dropout ratio |     |     |     |     |     |
|---------|---------------|-----|-----|-----|-----|-----|
|         | 0  | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| 1       | 0.759 | 0.474 | 0.308 | 0.109 | 0.000 | 0.000 |
| 3       | 0.553 | 0.527 | 0.521 | 0.479 | 0.439 | 0.445 |
| 4       | 0.511 | 0.502 | 0.482 | 0.470 | 0.483 | 0.465 |
| 5       | 0.942 | 0.600 | 0.259 | 0.152 | 0.000 | 0.000 |
| 6       | 0.655 | 0.619 | 0.551 | 0.513 | 0.459 | 0.354 |
| 7       | 0.614 | 0.590 | 0.554 | 0.551 | 0.494 | 0.443 |
| 8       | 0.482 | 0.485 | 0.477 | 0.478 | 0.451 | 0.459 |
| 9       | 0.282 | 0.278 | 0.285 | 0.288 | 0.278 | 0.300 |
| 10      | 0.432 | 0.430 | 0.432 | 0.395 | 0.383 | 0.375 |
| …      | … | … | … | … | … | … |
| 49      | 0.532 | 0.277 | 0.000 | 0.000 | 0.000 | 0.000 |
| 50      | 0.890 | 0.409 | 0.080 | 0.000 | 0.000 | 0.000 |
| 51      | 0.906 | 0.467 | 0.244 | 0.000 | 0.000 | 0.000 |
| 52      | 0.627 | 0.541 | 0.501 | 0.547 | 0.429 | 0.368 |
| Average | 0.668 | 0.452 | 0.314 | 0.240 | 0.182 | 0.155 |

Figure 6 shows the accuracy when the LSTM without dropout model was derived by randomly selecting 39 data from a total of 52 product time series data, and the test was performed using the remaining 13 product data. The LSTM without dropout model shows an accuracy of 66.8%.

---

**Figure 6.** LSTM model without dropout
4. Conclusion
The results of the study indicate that LSTM, which connect the present task to the previous data using neural network methods to predict the results, is a suitable technique for demand forecasting than simple exponential smoothing. Also, dropout was applied to prevent complex co-adaptation of training data and reduce overfitting in neural networks. As shown in Table 3 and Figure 6, no dropout leads to more accurate forecasting compared to the conventional LSTM method.

Dropout has been shown to be useful in the case of relatively large deep network models with a lot of training data. However, it is less effective in the case of models with sparse data. Therefore, there is a need to conduct research on the effectiveness of dropout algorithms under conditions of data sparsity.

Acknowledgments
This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2020-2018-0-01417) supervised by the IITP (Institute for Information & communications Technology Promotion).

References
[1] Sunyoung Cho. 2010 Time Series Prediction Using Recurrent Neural Network. Chungju University p 7-11
[2] Heeyeon Choi, Yunhong Min. 2015 Understanding Dropout Algorithms. Communications of the Korean Institute of Information Scientists and Engineers, 33(8), 32-38. p 1,4
[3] http://colah.github.io/posts/2015-08-Understanding-LSTMs/
[4] Samkeun Kim, Takil Oh. 2018 Real-time PM10 Concentration Prediction LSTM Model based on IoT Streaming Sensor data, Journal of the Korea Academia-Industrial cooperation Society, 19(11), p 310-318
[5] Sujin Lee. 2018 Deep Learning-based Estimation and Mapping of Evapotranspiration in Cropland using Local Weather Prediction Model and Satellite Data. The Korean Cartographic Association, 18(3), p 105-116.