ASM Sc. J., 13, Special Issue 1, 2020 for ICFICE, 82-91

Deep Learning Module Optimization based on Sequential Data Prediction

Jongwon Lee, Hongyoul Kim and Hoekyung Jung*

PaiChai University, 155-40 Baejae-ro, Seo-gu, Daejeon, Korea

The existing sequential data prediction systems are based on deep learning modules. Our study was based on observing past data and predicting data outcomes from the reference data. Although various methods have been studied for this purpose, there is insufficient research to optimize the setting values of the deep learning module. The proposed system consists of a pre-processor that refines an irregular data set into sequential data, and a deep learning module that performs learning and sequential data prediction. We investigate the fact that the accuracy depends on the dataset and, length of sequence, and shows the ratio of optimized batch size, dataset and the length of sequence. As a result, we can expect higher accuracy than the existing deep learning models based on sequential data prediction system. Using the proposed system, it is considered that the efficiency is higher than that of the existing systems in the field requiring less learning data and high accuracy.

Keywords: data refined; deep learning; DNN; machine learning; LSTM; pre-processing

I. INTRODUCTION

Data prediction systems use a Deep Neural Network (DNN) model to learn vast amount of data to perform predictions. The DNN model finds the rules of the data learned by the program and adapts it to the situation to derive appropriate results for the specific situation. Due to these structural features, data prediction systems were developed based on the DNN model (Li *et al.*, 2017; Jang *et al.*, 2017).

The DNN model is more accurate as the amount of learning data increases and the number of iterations of learning is larger. This means that the resources required to increase the accuracy of the program are large. If the amount of learning data is small or the number of iterations is less, the accuracy of the DNN model is low, which means that the efficiency is low. Also, there is a problem that a vanishing gradient phenomenon may occur (Lee & Nang, 2016; Ahn, 2016; Lee & Lee, 2016).

Another model used in data prediction systems to solve these drawbacks is the Recurrent Neural Network (RNN) model. The RNN model can be expected to have higher accuracy than the DNN model even though the weights used in the hidden layer of the program are changed according to the learning situation and the number of iterations is small. On the other hand, as the number of iterations increases, the RNN model increases the number of changes of the weights and increases the error range (Park, 2018; Kim *et al.*, 2016; Wang & Kim, 2018; Seo *et al.*, 2018). The program that predicts the data must be guaranteed to have high accuracy under any circumstances. Conditions such as the DNN model that the amount of learning data is large and the number of iterations of learning data should be large are not suitable for a program that predicts data. Also, it is not a suitable model even when the error range is large like the RNN model (Lee *et al.*, 2016; Shin *et al.*, 2017; Kim, 2017).

In this paper, we propose a system to refine an unstructured data set into sequential data in a pre-processor. It is a system that learns and predicts data to come out as specific data type. Depending on the dataset and length of sequence, the accuracy can be different. We perform a study on this to show the optimized batch size, the ratio of the dataset to the sequence length. In this paper, it is suggested that the reference index is wrong. As a result, it can be expected to have higher accuracy than

*Corresponding author's e-mail: hkjung@pcu.ac.kr

the conventional deep learning model-based data prediction system.

II. SYSTEM DESIGN

In this chapter, we have studied the setting values that affect the accuracy by DNN model, basic LSTM model, and stateful LSTM (Long Short-Term Memory) model. The DNN model has been studied for the number of hidden layers and layout size, and optimized values are derived and designed for implementation. The basic LSTM model performs a study on the batch size, and the stateful LSTM model performs a study on the weight initialization size. Figure 1 shows the configuration of the proposed system.

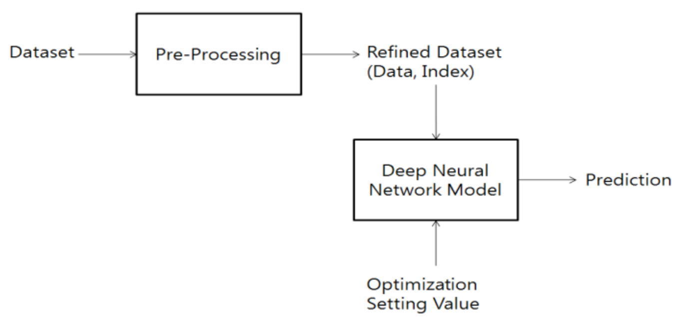


Figure 1. System Configuration

In the proposed system, a pre-processor performs a task of refining data set into a data: label value structure. The difference from the existing data refinement work is as follows. First, data refinement of existing system performs data refinement with 'data: label value' structure. Second, the data refining work of the proposed system performs data refinement with 'data: index' structure and 'index: data' structure. The reason for refining the data with the two structures is as follows. Third, the existing system updates the weight by the set batch size and proceeds to the data set with 'data: label value' structure. Fourth, the proposed system performs learning with sequential data set of 'data: index' structure and 'index: data' structure as many times as the optimized batch size. This is because the RNN model and the LSTM model are refined into sequential data sets of the data set, and then the accuracy is increased due to the structural characteristics of the model.

Existing systems have studied various methods for increasing the accuracy, but research on the ratio of the data set to the sequence length is insufficient. The proposed system analyses these existing researches and conducts research to solve them.

First, existing systems have mostly studied the number of hidden layers, and research on the ratio of data set to length of sequence is insufficient. When the sequence length is changed, the accuracy is different when the learning and data prediction are performed with the same data set. In order to improve the accuracy, it is necessary to carry out research and derive optimized values. Second, the proposed system performs the experiment by changing the ratio of DNN model-based system, basic LSTM model-based system, stateful LSTM model-based system dataset and length of sequence, and presents optimized values.

The system proposed in this paper uses DNN model, basic LSTM model, and stateful LSTM model among the deep learning models. Figure 2 shows the structure of the system.

In the pre-processor, the data input by the user is converted into a sequence of 'data: index' and 'index: data', and is refined as sequential data. Then, the sequential data that has been refined in the pre-processor is applied to the learning of the deep learning module. In the Deep Learning module, starting data to be applied to sequential data prediction is set, and sequence learning, batch size, and number of iterations is defined. When learning is complete, the system predicts the data and graphs the results to show the results to the user. This allows the user to know the accuracy of the functions provided by the proposed system. Figure 3 is a flow chart of the system.

When the system is started, the pre-processor converts the data entered by the user into two patterns. Data is converted into a structure of 'data: label value' type, and actual data is structure of 'data: index' and 'index: data'. The pre-processor also arranges the data into sequential data and uses it as learning data for the Deep Learning module. And performs learning based on weight updating as much as the optimized batch size. Figure 4 is a flow chart of the pre-processor.

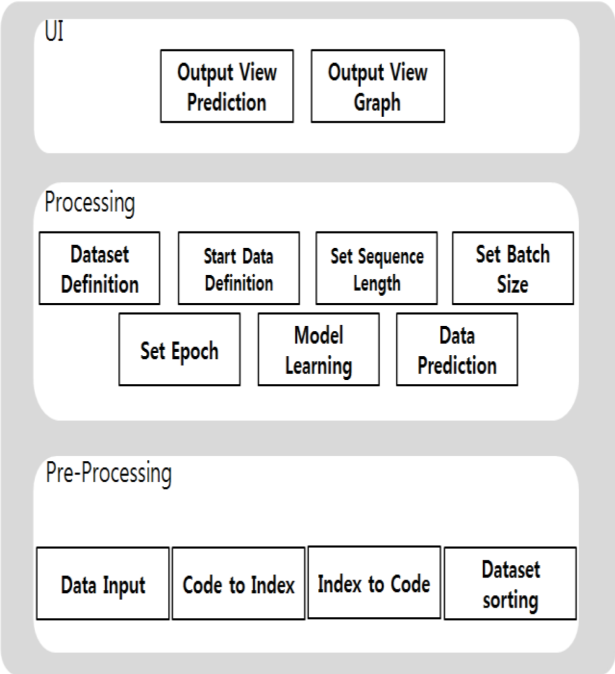


Figure 2. System Architecture

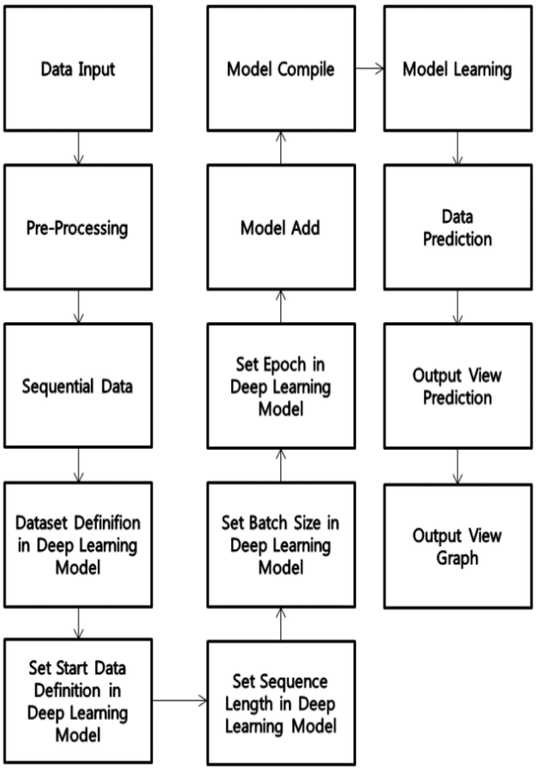


Figure 3. System Dataflow

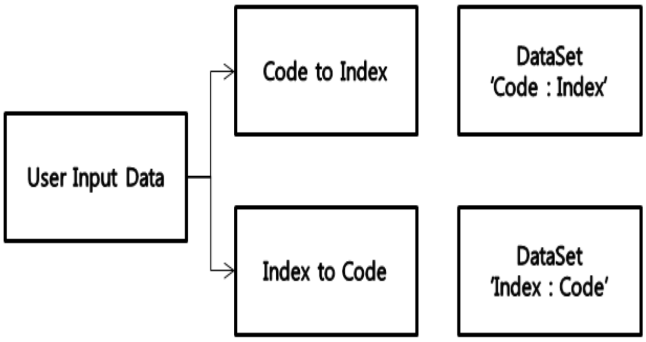


Figure 4. Pre-Processing Dataflow

The pre-processor refines the data set entered by the user into two types. In the conventional data prediction system, the refining work was performed in the form of giving the label value directly to the source code of the deep learning model. To prevent such a situation, the pre-processor executes the proposed system. In the pre-processor implemented in Java, the function is implemented based on the Scanner class and receives data from the user. The input data is refined into two patterns, both of which are derived from the 'data: label value' structure. Since the LSTM model performs learning with two values of data and index, two types of values are derived in this form.

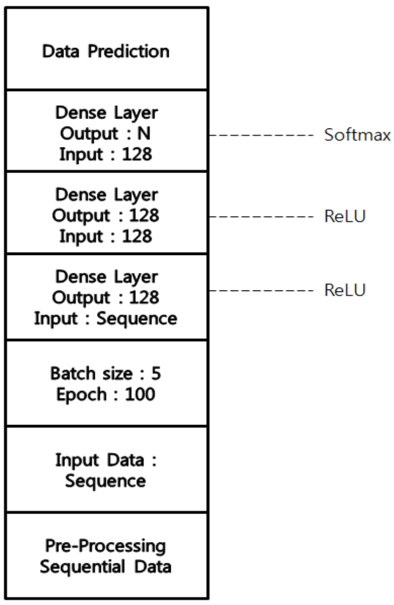


Figure 5. DNN Module based System Architecture

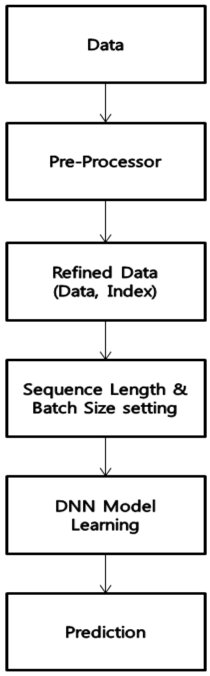


Figure 6. DNN Module based System Dataflow

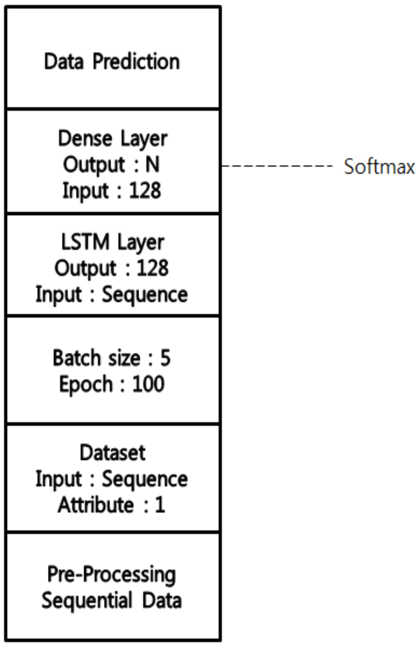


Figure 7. LSTM Module based System Dataflow

Figure 5 shows the structure of a DNN model-based data prediction system with three hidden layers designed for experiments, a batch size of 5, and number of iterations as 100. A sequential data having a length of 4 units is input and the next sequential data is assigned as a label value, and learning is performed based on the data of such structure.

Figure 6 shows the dataflow of a DNN module-based data prediction system. This begins with the user preparing the data and then performing the refinement. Then, the data set composed of the sequential data that has been refined is applied to the DNN model to perform learning. The first and second hidden layers that perform learning use ReLU as an activation function. The third hidden layer uses Softmax as an active function. When the learning is completed, the user performs a task of predicting data to be used as next input for the input data.

Figure 7 shows the structure of a basic LSTM model-based data prediction system in which the data set 52, the sequence length 4, the number of attributes 1, the batch size 5, and the number of iterations is set to 100. A sequential data having length of 4 units is input and the next sequential data is assigned as a label value, and learning is performed based on the data of such structure.

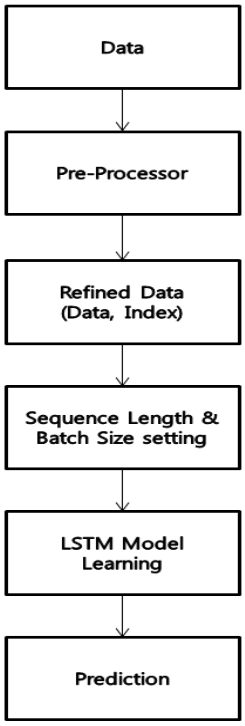


Figure 8. LSTM Module based System Dataflow

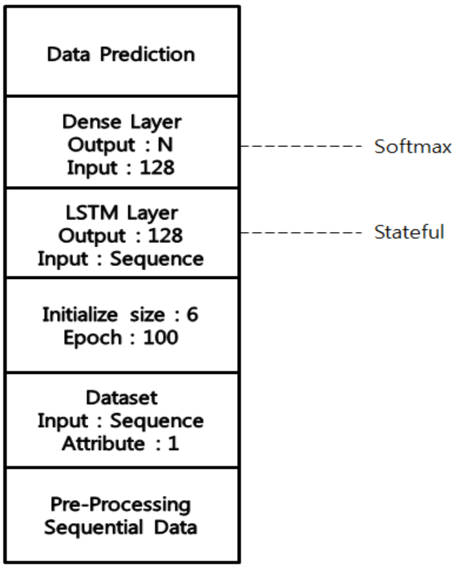


Figure 9. stateful LSTM Module based System Architecture

Figure 8 shows the dataflow of a basic LSTM model-based data prediction system which begins with the user preparing the data and then performing the refinement. Subsequently, the data set composed of the sequential data that has been refined is applied to the deep learning model to perform learning. In learning, we use the active function Softmax in the third hidden layer of the DNN model. When the learning is completed, the user performs a task of predicting data to be used as next input for the input data.

Figure 9 shows the structure of a stateful LSTM model-based data prediction system with a data set 52, a sequence length of 4, an attribute number of 1, a weight initialization size of 6, and number of iterations of 100. Sequential data composed of 4 units in length serves as input and the next sequential data is assigned as a label value. The learning state is memorized and influenced in the next learning. The results are different depending on whether the state is maintained or not. The maintenance of the state in the experiment is carried out by initializing the weight at the start of a new learning.

Figure 10 shows dataflow of a stateful LSTM model-based data prediction system which begins with the user preparing the data and then performing the refinement. Subsequently, the refined data set is applied to the deep learning model to perform learning. In learning, we use Softmax as an active function used in the third hidden layer of the DNN model and the basic LSTM model. In addition, previous learning through the stateful function influences the next learning as the learning progresses. When the learning is completed, the user performs a task of predicting data to be used as next input for input data.

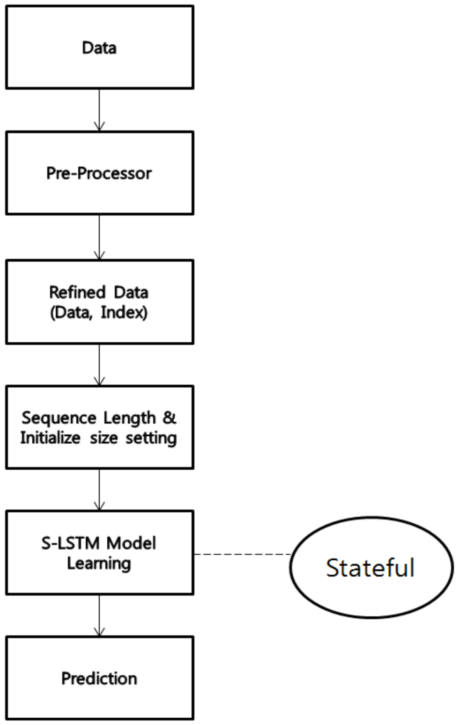


Figure 10. stateful LSTM Module based System Dataflow

III. SYSTEM MPLEMENTATION AND REVIEW

The PC to be implemented and tested uses CPU Core (TM) i5-4690 3.50GHz, RAM 8G, and OS Windows 7 64bit. The tools used were ‘Anaconda 3’ and ‘jupyter notebook’, ‘Eclipse’. They tools used implement to deep learning models and pre-processor. Figure 11 and Figure 12 shows the output screen of the pre-processor.

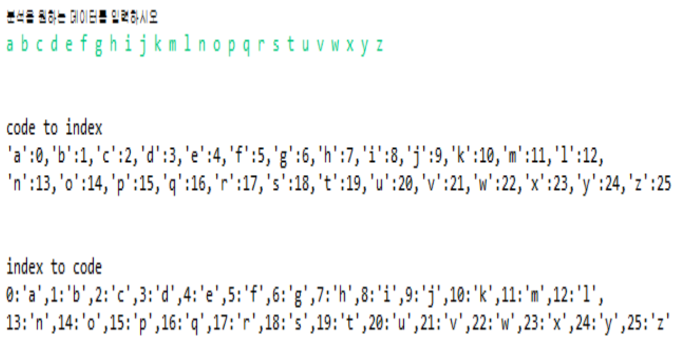


Figure 11. Screen of Pre-processing

```
분석을 원하는 데이터를 입력하시오
My Heart is like a singing bird Whose nest is in a watered shoot


code to index
'My':0,'Heart':1,'is':2,'like':3,'a':4,'singing':5,'bird':6,
'Whose':7,'nest':8,'is':9,'in':10,'a':11,'watered':12,'shoot':13


index to code
0:'My',1:'Heart',2:'is',3:'like',4:'a',5:'singing',6:'bird',
7:'Whose',8:'nest',9:'is',10:'in',11:'a',12:'watered',13:'shoot'
```

Figure 12. Screen of Pre-processing 2

Data is indexed and pre-processed with 'data: index' structure and 'index: data' structure. And the data refined in the pre-processor is used for the learning of the deep learning model as shown in Figure 13 and the learning process of the deep learning model is shown in Figure 14.

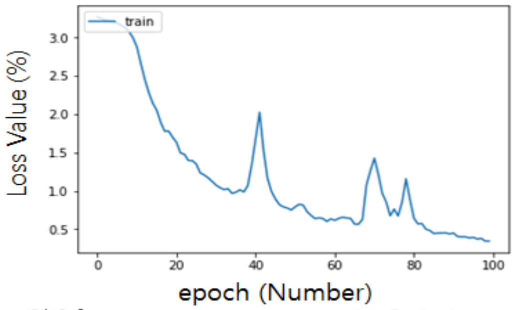
```
(39, 14)
[[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13]
 [ 1  2  3  4  5  6  7  8  9 10 11 12 13 14]
 [ 2  3  4  5  6  7  8  9 10 11 12 13 14 15]
 [ 3  4  5  6  7  8  9 10 11 12 13 14 15 16]
 [ 4  5  6  7  8  9 10 11 12 13 14 15 16 17]
 [ 5  6  7  8  9 10 11 12 13 14 15 16 17 18]
 [ 6  7  8  9 10 11 12 13 14 15 16 17 18 19]
 [ 7  8  9 10 11 12 13 14 15 16 17 18 19 20]
 [ 8  9 10 11 12 13 14 15 16 17 18 19 20 21]
 [ 9 10 11 12 13 14 15 16 17 18 19 20 21 22]
 [10 11 12 13 14 15 16 17 18 19 20 21 22 23]
 [11 12 13 14 15 16 17 18 19 20 21 22 23 24]
 [12 13 14 15 16 17 18 19 20 21 22 23 24 25]
 [13 14 15 16 17 18 19 20 21 22 23 24 25  0]
 [14 15 16 17 18 19 20 21 22 23 24 25  0  1]
 [15 16 17 18 19 20 21 22 23 24 25  0  1  2]
 [16 17 18 19 20 21 22 23 24 25  0  1  2  3]
 [17 18 19 20 21 22 23 24 25  0  1  2  3  4]
 [18 19 20 21 22 23 24 25  0  1  2  3  4  5]
 [19 20 21 22 23 24 25  0  1  2  3  4  5  6]
 [20 21 22 23 24 25  0  1  2  3  4  5  6  7]
 [21 22 23 24 25  0  1  2  3  4  5  6  7  8]
 [22 23 24 25  0  1  2  3  4  5  6  7  8  9]
 [23 24 25  0  1  2  3  4  5  6  7  8  9 10]
 [24 25  0  1  2  3  4  5  6  7  8  9 10 11]
 [25  0  1  2  3  4  5  6  7  8  9 10 11 12]
 [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13]
 [ 1  2  3  4  5  6  7  8  9 10 11 12 13 14]
 [ 2  3  4  5  6  7  8  9 10 11 12 13 14 15]
 [ 3  4  5  6  7  8  9 10 11 12 13 14 15 16]
 [ 4  5  6  7  8  9 10 11 12 13 14 15 16 17]
 [ 5  6  7  8  9 10 11 12 13 14 15 16 17 18]
 [ 6  7  8  9 10 11 12 13 14 15 16 17 18 19]
 [ 7  8  9 10 11 12 13 14 15 16 17 18 19 20]
 [ 8  9 10 11 12 13 14 15 16 17 18 19 20 21]
 [ 9 10 11 12 13 14 15 16 17 18 19 20 21 22]
 [10 11 12 13 14 15 16 17 18 19 20 21 22 23]
 [11 12 13 14 15 16 17 18 19 20 21 22 23 24]
 [12 13 14 15 16 17 18 19 20 21 22 23 24 25]]
```

Figure 13. Screen of Dataset Stucture

```
Epoch 80/100
 - 0s - loss: 0.6949 - acc: 0.8095
Epoch 81/100
 - 0s - loss: 0.6803 - acc: 0.7619
Epoch 82/100
 - 0s - loss: 0.7671 - acc: 0.5476
Epoch 83/100
 - 0s - loss: 1.0452 - acc: 0.5238
Epoch 84/100
 - 0s - loss: 0.8223 - acc: 0.6429
Epoch 85/100
 - 0s - loss: 0.8392 - acc: 0.6190
Epoch 86/100
 - 0s - loss: 0.6328 - acc: 0.8571
Epoch 87/100
 - 0s - loss: 0.5462 - acc: 0.8810
Epoch 88/100
 - 0s - loss: 0.5523 - acc: 0.9286
Epoch 89/100
 - 0s - loss: 0.5370 - acc: 0.8810
Epoch 90/100
 - 0s - loss: 0.5153 - acc: 0.9524
Epoch 91/100
 - 0s - loss: 0.4608 - acc: 0.9762
Epoch 92/100
 - 0s - loss: 0.4567 - acc: 0.9048
Epoch 93/100
 - 0s - loss: 0.4442 - acc: 1.0000
Epoch 94/100
 - 0s - loss: 0.4246 - acc: 1.0000
Epoch 95/100
 - 0s - loss: 0.4214 - acc: 0.9762
Epoch 96/100
 - 0s - loss: 0.4141 - acc: 0.9286
Epoch 97/100
 - 0s - loss: 0.4095 - acc: 0.9762
Epoch 98/100
 - 0s - loss: 0.4119 - acc: 0.9762
Epoch 99/100
 - 0s - loss: 0.3986 - acc: 0.9524
Epoch 100/100
 - 0s - loss: 0.3790 - acc: 0.9762
```

Figure 14. Screen of Training

Figure 15 is a graph showing the accuracy of the Deep Learning module when the learning progresses and the sequential data prediction are performed based on the learning.




```
43/43 [==============================] - 0s 4ms/step
acc: 100.00%
```

Figure 15. Screen of Training Result and Prediction

Experiments are carried out by changing data set and sequence length of systems implemented on the basis of Deep Learning Module, and the accuracy of each module is analysis to compare the average accuracy and the time required for learning and data prediction. Then, the most suitable system is selected through the comparative analysis. In this experiment, two data sets consisting of 28 data and 52 data are inserted and the data is predicted. When learning through data set 28, proceed with the experiment by increasing the sequence length from 2 to 14. When learning through the data set 52, the sequence length is increased from 2 to 26 and the experiment is continued. Based on the results of 38 experiments for each model-based system, an optimized ratio of dataset and length of sequence is presented.

The conclusions drawn from experiments performed with the DNN model - based data prediction system are as follows. The accuracy reached 100 percent when the ratio of data length to the sequence length was 2 : 1 in the dataset. This resulted in high accuracy even though the number of iterations was 100 times. Based on these results, the ideal setting of DNN model based data prediction system is batch size 5, and the ratio of dataset to sequence length is 2 : 1. Average accuracy was the lowest among the three modules.

The conclusions drawn from experiments performed with the basic LSTM model-based data prediction system are as follows. If the dataset is small, the accuracy reaches 100 percent. On the other hand, if the amount of data set increases, the accuracy decreases slightly. The closer the ratio of data set to the sequence length, the closer the accuracy. Based on these results, the ideal setting of the basic LSTM model based data prediction system is batch size 5, and the ratio of data set to sequence length is 6 : 1. The average accuracy was medium among the three models.

State LSTM model-based data prediction system. The conclusions drawn from the experiments are as follows. In the experiments performed with Dataset 28, the average accuracy was lower than that of the basic LSTM model-based data predicting system, but it was higher in the experiments performed with dataset 52. This is because the weight initialization method of the stateful LSTM model is different from other models, so that the efficiency increases as the number of data sets increases. Also, from 5 : 1 ratio of dataset to the sequence length, 100% accuracy was obtained regardless of the sequence length changes. Based on these results, the ideal setting value of the LSTM model based data prediction system is weight initialization size 6, and the ratio of data set to sequence length is 5 : 1. Average accuracy was the highest among the three modules. Figure 16 shows the results of the accuracy test of Deep Learning Modules.

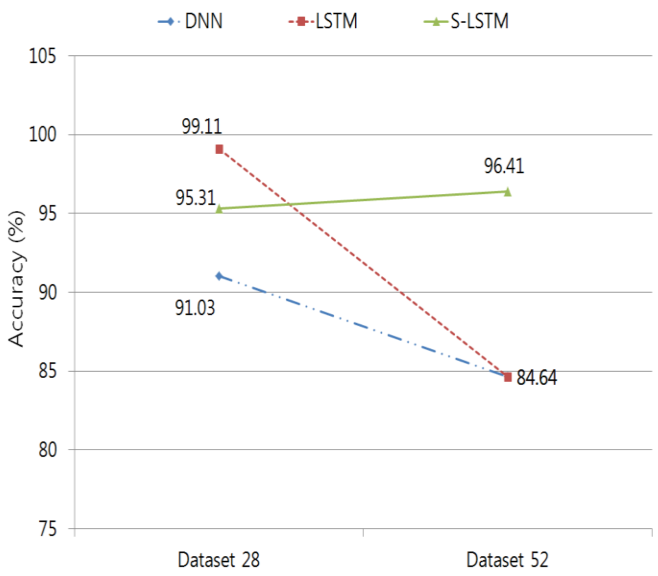


Figure 16. Accuracy of Three Models

Table 1 summarizes the time required for learning and data prediction. Table 3 shows the result of comparison analysis with existing system.

Table 1. Time for Training and Prediction

Dataset	DNN Model	LSTM Model	S – LSTM Model
Dataset 28	1ms ~ 2ms	3ms ~ 4ms	2s ~ 3s
Dataset 52	2ms ~ 3ms	3ms ~ 4ms	3s ~ 4s

DNN model-based data prediction system has the lowest accuracy but has the shortest time for learning and data prediction. The basic LSTM model-based data prediction system has the highest accuracy in dataset 28 and much less accuracy in dataset 52. Nevertheless, the error range was the narrowest. The LSTM model-based data prediction system has the highest average accuracy but takes a long time for learning and data prediction. Also, the longer the data set, the more time it takes to exponentially increase the efficiency of the system if it is not built with a high-performance computer.

The basic LSTM model-based system is the most efficient if each system is evaluated considering the accuracy and the time required for data prediction. State LSTM model-based data prediction system can be applied to basic LSTM model-based systems at similar times because the time required is high even if the accuracy is high. The number of iterations can be increased, thereby increasing the accuracy and maintaining the state. State LSTM model-based data prediction system spent over 10 times longer in learning and data prediction than the basic LSTM model-based data prediction system. This is longer than the time required by the basic LSTM model-based data prediction system in which the number of iterations is set to 1000 times or more. In addition, the basic LSTM model-based data prediction system has reached an average accuracy of 100 percent since the number of iterations is 200 or more. Based on these results, the best model for data prediction is the basic LSTM model.

The proposed system analyses existing data prediction systems and developed a pre-processor to improve accuracy. The pre-processor refines the unstructured data into sequential data so that the deep learning model can learn. And Deep Learning model is based on three modules.

Table 2. Comparisons of Three Models

Attribute	DNN Model	LSTM Model	S–LSTM Model
Average Accuracy	Rank 3	Rank 2	Rank 1
Training Speed	Rank 1	Rank 2	Rank 3
Efficiency	Rank 2	Rank 1	Rank 3
Batch Size	5	5	1
Weight Update	0	0	6
Ratio	2 : 1	6 : 1	5 : 1

Table 2 summarizes the results of performance evaluation of each model. The basic LSTM model-based system is most efficient if we evaluate each system considering the accuracy and the time required for data prediction. State LSTM model-based data prediction system, even if the accuracy is high, the time required is long. Therefore, if similar time is applied to the basic LSTM model-based system, the number of iterations can be increased. State LSTM model-based data prediction system took 10 times longer in learning and data prediction than the basic LSTM model-based data prediction system. This is longer than the time required by the basic LSTM model-based data prediction system, which sets the number of iterations to 1000 or more times. In addition, the basic LSTM model-based data prediction system reached an average accuracy of 100 percent since the number of iterations was 200 or more. Based on these results, the best model for data prediction is the basic LSTM model.

Table 3. Comparisons Analysis

Attribute	Existing System	Proposed System
Dataset	Non-sequential Data	Sequential Data
Using Model	DNN, RNN, LSTM	LSTM
Optimization Setting Value	Impossible	Possible
Ratio Setting	Impossible	Possible
Accuracy	70 ~ 90	90 ↑
Epochs	2000 ↑	100

Existing systems mainly use non - sequential data for learning, and the proposed system uses sequential data for learning. Because of the structural characteristics of the LSTM model, the accuracy improves when learning based on sequential data, so we performed the process of refining unstructured data into sequential data through pre-processor. We used various deep learning models such as DNN model, RNN model, and LSTM model as the usage models of existing systems. In this paper, we have constructed a system based on DNN model, basic LSTM model, and state-preserving LSTM model and confirmed that the basic LSTM model-based system is the most efficient through performance comparison experiments.

Existing systems focused on setting value optimization as the number of hidden layers or the amount of datasets. In contrast to this paper, we have analysed the batch size, the ratio of the data set to the length of sequence and confirmed that learning and data prediction based on the same data set and model can improve the accuracy by optimizing the set value. The accuracy of existing systems was typically between 70 percent and 90 percent. On the other hand, the proposed system showed an average accuracy of 91.87 percent. This improves the accuracy compared to the existing systems. Existing systems generally performed over 2000 iterations. On the other hand, the proposed system has higher accuracy than the existing system and it takes less time to predict the data even though the number of iterative learning is set to 100 and the learning and data prediction are performed. The following conclusions were drawn through the analysis of these experimental results;

■ We used refining data for learning then improved accuracy.

■ Basic LSTM model is suitable for data prediction system.

■ Batch size, ratio of dataset to sequence length, etc can improve accuracy and reduce learning and data prediction time.

In this paper, we refined the data to be used for learning into sequential data, and improved the accuracy of prediction and reduced the time required to optimize the set value of the basic LSTM model. This proves that the proposed system is more accurate and efficient than existing systems. When the system is constructed for data prediction, it can be expected to improve performances as compared to the existing systems by using the proposed values and the pre-processor.

IV. CONCLUSIONS

Systems for predicting data are being developed using various methods. Most of the systems that use the deep learning model to predict the data that follows the specific data have been developed based on the DNN model. The DNN model sets the length of the sequence, the number of iterations during the data set and learning process. Then, the model predicts.

In this paper, we developed a data prediction system based on preprocessor and deep learning module. In the preprocessor, the deep learning module refines the irregular data into sequential data of "data: index" structure before performing the learning. Experiments were carried out on the set values of the Deep Learning module and the optimized batch size and weight initialization size were derived. In addition, we confirmed that the accuracy varies according to the ratio of dataset and sequence length, and analyzed the results to show the degree of accuracy of the deep learning module. Based on this optimization study, we implemented the Deep Learning module based sequential data prediction system and analyzed through experiments. The most suitable Deep Learning module is the basic LSTM model. Therefore, if the system is constructed for data prediction, it is effective to build a system based on the preprocessor and the basic LSTM model to refine the data with sequential data. And it can be expected that high accuracy can be obtained by changing the setting value of the model.

V. ACKNOWLEDGEMENT

This work was supported by the research grant of Pai Chai University in 2019.

VI. REFEREENCES

D.H. Seo, J.S. Lyu, E.J. Choi, S.H. Cho, and D.K. Kim 2018, 'Web based Customer Power Demand Variation Estimation System using LSTM', *Journal of the Korea Institute of Information and Communication Engineering*, vol. 22, no. 4, pp. 587-594.

D.H. Shin, K.H. Choi, C.B. Kim 2017, 'Deep Learning Model for Prediction Rate Improvement of Stock Price Using RNN and LSTM', *The Journal of Korean Institute of Information Technology*, vol. 15, no. 10, pp. 9-16.

I.S. Jang, C.H. Ahn, J.G. Seo, and Y.S. Jang 2017, 'DNN based Speech Detection for the Media Audio', *Journal of Broadcast Engineering*, vol. 22, no. 5, pp. 632-642.

J.Li, S.Y. Shin, and H.C. Lee 2017, 'Text Mining and Visualization of Papers Reviews Using R Language', *Journal of Information and Communication Convergence Engineering*, vol. 15, issue. 3, pp. 170-174.

J.Y. Park 2018, 'Estimation of Electrical Loads Patterns by Usage in the Urban Railway Station by RNN', *The Transactions of the Korean Institute of Electrical Engineers*, vol. 67, no. 11, pp. 1536-1541.

S.H. Lee, and J.H. Lee 2016, 'Customer Churn Prediction Using RNN', *Proceedings of the Korean Society of Computer Information Conference*, vol. 24, no. 2, pp. 45-48.

S.M. Ahn 2016, 'Deep learning architectures and applications', *Journal of Intelligence and Information Systems*, vol. 22, no. 2, pp. 127-142.

W.K. Lee, S.Y. Ahn, M.S. Yim, S.T. Chun 2016, 'Forecasting Energy Data Using LSTM', *Journal of Korea Information and Science Society*, pp. 693-695.

X.F. Wang, and H.C. Kim 2018, 'Text Categorization with Improved Deep Learning Methods', *Journal of Information and Communication Convergence*

Engineering, vol. 16, no. 2, pp. 106-113.

Y.H. Kim 2017, 'Distributed Estimation Using Non-regular Quantized Data', *Journal of Information and Communication Convergence Engineering*, vol. 15, no. 1, pp. 7-13.

Y.H. Kim, Y.K. Hwang, T.G. kang, and K.M. Jung 2016, 'LSTM Language Model Based Korean Sentence Generation', *The Journal of Korean Institute of Communications and Information Sciences*, vol. 41, no. 5, pp. 592-601.

Y.J. Lee, and J.H. Nang 2016, 'A Personal Video Event Classification Method based on Multi-Modalities by DNN-Learning', *Journal of Korea Information Science Society*, vol. 43, no. 11, pp. 1281-1297.