

Wind Speed Prediction Model Using LSTM and 1D-CNN

Rui Fukuoka, Hiroshi Suzuki, Takahiro Kitajima, Akinobu Kuwahara and Takashi Yasuno

Graduate School of Advanced Technology and Science, Tokushima University
2-1 Minami-Josanjima, Tokushima 770-8506, Japan
E-mail: {fukuoka-r, kitajima, akinobu}@ee.tokushima-u.ac.jp
{suzuki.hiroshi, yasuno.takashi}@tokushima-u.ac.jp

Abstract

This paper describes a prediction method for wind speed using a neural network and an investigation of the structure of the network. Generally, wind speed is observed as time series data, and the current wind speed is related to the past wind speed. Therefore, we propose a prediction model using long short-term memory (LSTM) and a one-dimensional convolutional neural network (1D-CNN) in order to consider the past information for prediction. The prediction results of these networks and a fully connected neural network are compared for evaluation. The prediction accuracy and time delay are found to be improved by using LSTM and the 1D-CNN.

1. Introduction

Currently, most electricity is generated from fossil fuels. However, the amount of fossil fuels is limited, and CO₂, which is a cause of global warming, is emitted during power generation. Therefore, in recent years, renewable energy has attracted attention because it does not emit CO₂ during power generation. Among the various types of renewable energy, wind power generation is being introduced rapidly all over the world since it can generate electricity continuously if wind blows. However, wind power generation has a problem of output fluctuation due to wind speed changes. As measures against this problem, electric companies regulate the supply of power using thermal power generation and pumped-storage hydroelectricity. Hence, it is necessary to predict the output of wind power generation to regulate the supply of power effectively.

So far, our research group has proposed several types of wind speed prediction system using a neural network. However, there is a time-delay prediction error[1]. In this paper, we propose a new wind speed prediction system with long short-term memory (LSTM) and a one-dimensional convolutional neural network (1D-CNN), both of which are effective for considering time-series data. LSTM is an extension of a recurrent neural network (RNN), which inputs the unit's output to the same unit recursively. It can learn the long-term dependence of time series data, and it is often used for sentence generation and dialogue system. A 1D-CNN can ex-

tract features by convolving time series data in the time direction, and it is used for sound recognition. Focusing on these features, LSTM and a 1D-CNN are expected to be effective methods for wind speed prediction. The usefulness of these networks is evaluated from the comparison of prediction results in terms of the accuracy and time delay of prediction outputs. Here, we use a fully connected neural network for comparison with these networks.

2. Wind Speed Prediction System

2.1 Prediction period and input data

The prediction period is set to 1 h later in consideration of the adjustment ability of thermal power generation and pumped-storage hydroelectricity. We use the wind speed data of Tokushima city acquired from AMeDAS at 10 min intervals as input and teaching data. However, it is difficult to make predictions from only the wind speed. Therefore, we used the wind direction and extraterrestrial solar radiation as input data, which include information on seasonality, regionality, and the time of day. These are also data for Tokushima city given at 10 min intervals. The wind direction acquired from AMeDAS is expressed as one of 16 directions. This is used to decompose the wind speed into north-south and east-west components. The extraterrestrial solar radiation is the theoretical value. An example of input data is shown in Fig.1; data is normalized to values from 0 to 1 for inputting because each set of data has a different scale.

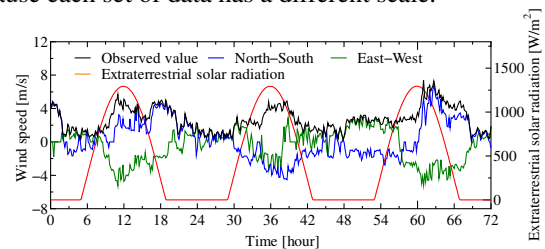


Figure 1: Input data for May 20-22, 2016 at Tokushima

2.2 Constitution of fully connected neural network

Figure 2 shows the fully connected neural network, which consists of an input layer, three middle layers, and an output layer. In order to consider past information, the input

is data for one day (144 dimensions) including the current value. Each layer is connected with weight $w_{ij}^{(k)}$ and has one bias weight $b_i^{(k)}$. The output from the unit is calculated by the activation function ReLU ($\phi(u) = \max(0, u)$) from internal state u . Hence, the i th output of the k th middle layer is expressed by

$$h_i^{(k)} = \phi \left(\sum_{j=1}^{M^{(k)}} w_{ij}^{(k)} x_j^{(k)} + b_i^{(k)} \right) \quad (1)$$

where $M^{(k)}$ is the number of inputs and $x_j^{(k)}$ is the value input to the k th middle layer. The output is the input of the next layer. The weights and biases of each unit are adjusted by the backpropagation (BP) method. Here, the loss function is the squared error.

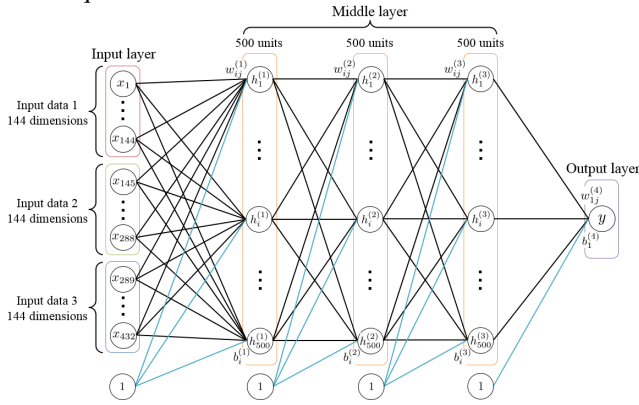


Figure 2: Constitution of fully connected neural network

2.3 Constitution of LSTM

Figure 3 shows the constitution of a layer of the LSTM[2]. Here, inputs, outputs, and weights are represented by vectors and matrices, and Fig.3 shows the operation of a layer. The LSTM block has an input gate, output gate, and forget gate[3]. These gates realize learning long-term dependences. The model using the LSTM is constructed by replacing units of middle layers of the fully connected neural network in Fig.2 with LSTM blocks. Figure 4 shows the LSTM unfolded in the time axis direction. Weights and biases are adjusted by

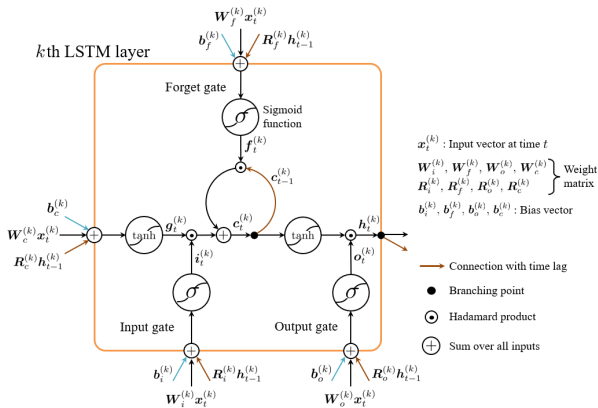


Figure 3: Constitution of LSTM layer

backpropagation through time (BPTT), where the loss function is the squared error. BPTT can update weights in consideration of a time series because it backpropagates the error to the recursive connection. However, when the time series becomes longer, the BP route also becomes longer. This can be regarded as the layer becoming deep, and the calculation cost increases. Therefore, in order to reduce the calculation cost, we use truncated BPTT. This backpropagates the error only in a certain period, such as for the sequence length in Fig.4, during training. Here, we set the sequence length to 144, which is the same as the input length of a fully connected neural network and the number of data for one day.

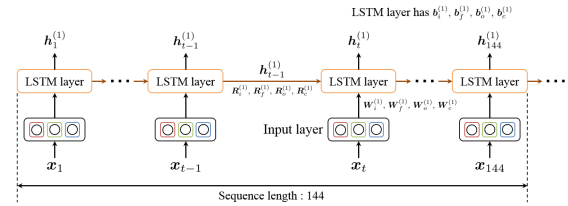


Figure 4: Operation of LSTM

2.4 Constitution of 1D-CNN

Figure 5 shows the constitution of the 1D-CNN. In order to consider past information, the input is data for one day (144 dimensions) including the current value. If there are multiple input data, the input vectors are superimposed in the channel direction as shown in Fig.5. In Fig.5, $C^{(k)}$ is the number of channels in the k th convolutional layer and H is the size of the filter. The activation function is ReLU. Therefore, the convolution process of the i th filter is expressed by

$$h_{ij}^{(k)} = \phi \left(\sum_{p=1}^{C^{(k-1)}} \sum_{q=1}^H w_{ipq}^{(k)} x_{p,j+q}^{(k)} + b_i^{(k)} \right) \quad (2)$$

where $w_{ipq}^{(k)}$ is the weight of the filter, $b_i^{(k)}$ is the bias of the filter, and $x^{(k)}$ is input to the k th layer. The number of filters in each layer is 16, 32, and 64, and H is set to 3. The convolution processes are performed while sliding the filter with a stride width of 1. After passing through the three convolutional layers, the created feature map is input to the fully connected output layer. The weight and bias are updated by BP. Here the loss function is the squared error.

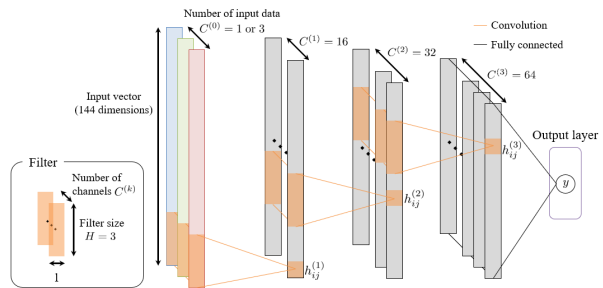


Figure 5: Constitution of 1D-CNN

3. Prediction Results

In this study, all prediction models are implemented by Chainer[4] and trained with data from 2010 to 2014. The optimization function is Adam. The training method is mini-batch training and the batch size is 100. Gradient clipping with a threshold of 5 is performed to prevent gradient explosion. Table 1 shows the input datasets of two patterns, Type1 and Type2, used to consider the effectiveness of increasing the input data. All prediction models are evaluated for loss transition in the training process from the prediction result for 2015, and the best epoch is decided from the transition of loss. We evaluate the prediction result for 2016. The prediction accuracy is evaluated by the root-mean-square error (RMSE) of the predicted wind speed, and the time delay is evaluated by the cross-correlation function. The data of the first day is only used for the input in order to align the prediction period of all models.

Table 1: Types of input datasets

	Input data
Type1	Wind speed (no normalization)
Type2	North-south wind speed East-west wind speed Extraterrestrial solar radiation

3.1 Loss transition in training process

Figures 6(a) and 6(b) show the transition of loss in the training process in each network. The loss for the training data decreases in all models. In contrast, the loss for the validation data starts to increase from the early stage of training. Hence, after 500 epochs, it can be seen that overtraining occurs in all cases. In the case of LSTM, the loss for the training data fluctuates, the loss at 500 epochs is the smallest, and the output fits the training data very well as shown in Fig.6(a). We consider that these features show the effectiveness of the LSTM.

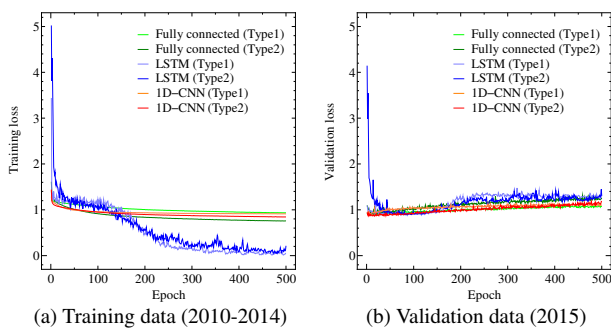


Figure 6: Transition of loss in training process

3.2 Prediction error and time delay

In the evaluation of the prediction models, we use the weights of the epoch where the loss for the validation data is nearly minimum for each model as shown in Table 2. We predict the data for 2016 and evaluate the results. Table 3

shows the RMSE of the prediction for 2016. Comparing the network types, the LSTM has the smallest RMSE in both Type1 and Type2, and the prediction accuracy is good. On the other hand, comparing the input type, the RMSE of Type2 is smaller, except in the case of a fully connected neural network. In particular, the prediction accuracy of the 1D-CNN is improved. It is found that the 1D-CNN is valid in the case of multiple input data.

Table 2: Epoch for evaluation of RMSE

Input data	Fully connected	LSTM	1D-CNN
Type1	24	86	2
Type2	31	86	17

Table 3: Prediction RMSE [m/s]

Input data	Fully connected	LSTM	1D-CNN
Type1	0.939	0.938	0.940
Type2	0.967	0.918	0.918

In order to evaluate the time delay of the prediction result, the cross-correlation function between the calculated value and the prediction result is obtained as shown in Fig.7. The cross-correlation function is defined by

$$z(\tau) = \frac{1}{N} \sum_{t=0}^{N-1} \hat{v}(t)v(t+\tau) \quad (3)$$

where v is the observed value, \hat{v} is the predicted value, τ is the time lag, and N is the number of data. In Fig.7 the persistent model presupposes that the current wind speed will be sustained until the prediction time (1 h). Hence, the prediction of the persistent model is the waveform of the observed value with a 1 h delay, and the cross correlation has a peak at 1 h. If the prediction result becomes accurate and decreases the time delay, the peak of the cross correlation will move from 1 h to 0 h. However, in Fig.7, all the peaks for the prediction results are located at 1 h, indicating that the models cannot reduce the time delay further. Therefore, we discuss the degree of the improvement of the time delay from the viewpoint of the ratio of the cross correlation at 1 h and 0 h. The ratio of the

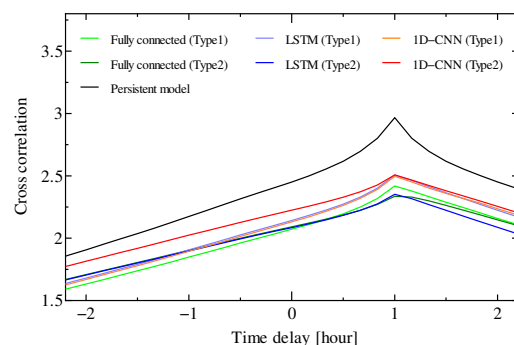


Figure 7: Cross correlation between observed value and prediction value

cross correlation to each prediction result is shown in Table 4. For the persistent model, the ratio is 0.826. By comparing the input type, it is found that the ratio of Type2 is larger than that of Type1. Therefore, the time delay is reduced by adding the wind direction and extraterrestrial solar radiation to the input. On the other hand, by comparing the models, it was found that there is not much difference in the results.

Table 4: Ratio of cross correlation

Input data	Fully connected	LSTM	1D-CNN
Type1	0.856	0.855	0.853
Type2	0.895	0.886	0.887

To consider the prediction result for each day, the prediction output of May 21, 2016 is shown in Figs.8(a) and 8(b). By comparing the input type, it can be confirmed that the time delay is improved around 16:00. Also, the predicted values of the LSTM and 1D-CNN are close to the observed values in Fig.8(b).

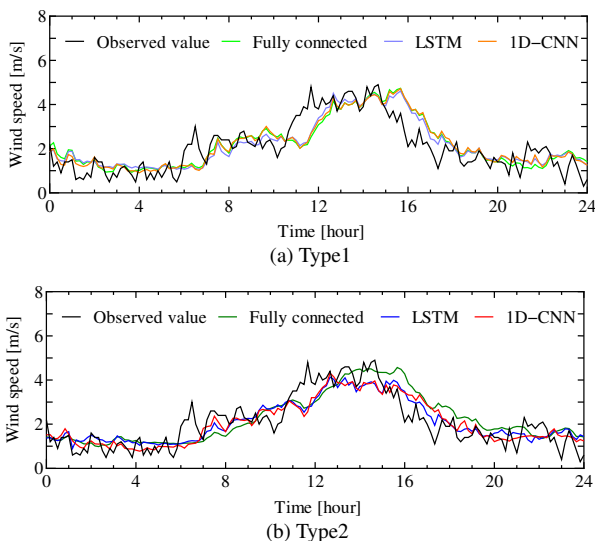


Figure 8: Prediction result for May 21, 2016 at Tokushima

As a result, the LSTM and 1D-CNN with the Type2 input have good prediction accuracy and the time delay is reduced. On the other hand, even if Type2 is used for the fully connected neural network, the prediction accuracy is not improved. This is thought to be related to the number of dimensions of the input. In the case of the fully connected neural network, the number of dimensions of the input is extremely large, 432, in consideration of the past day of each of the three inputs. On the other hand, the LSTM has a structure that can consider time series; thus, it that inputs only the current value, and the number of dimensions is 3. In the case of the 1D-CNN, the number of dimensions of the input is large and the same as that of the fully connected neural network. However, the 1D-CNN has good prediction accuracy. In the case of increasing the input data, the input of the 1D-CNN is added in the channel direction and the input data is

convolved by the filter. Hence, time information is not lost. The fully connected neural network cannot process time series data and timing information between the three inputs effectively because of the structure of the connection. Since the 1D-CNN can extract features from the input with time information even if the input data increases, the prediction accuracy is good. The significance of adding an input that assists prediction is indicated by the prediction results. Therefore, the LSTM and 1D-CNN can cope with an increase in input data while considering time series, making them effective for wind speed prediction. In particular, the LSTM can add a large number of input data because it has an input vector with fewer dimensions.

4. Conclusions

We predicted the 1-hour-later wind speed using a fully connected neural network, LSTM, and a 1D-CNN and compared their prediction results. Also, we compared the case of inputting only the wind speed and the case of inputting normalized data of the north-south/east-west wind speed and extraterrestrial solar radiation. The prediction accuracy and time delay were improved by using the north-south/east-west wind speed and extraterrestrial solar radiation for the LSTM and 1D-CNN. Also, the LSTM is expected to improve them by adding a large numbers of valid input data.

A future task is to consider valid input data is addition to the wind direction and extraterrestrial solar radiation. In addition, we will search for the optimum hyperparameters, such as the number of layers, units, and filters, for wind speed prediction by these methods.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 15K00048.

References

- [1] S. Yoshida, H. Suzuki, T. Kitajima, A. M. Kassim and T. Yasuno: Correction method of wind speed prediction system using predicted wind speed fluctuation, SICE Annual Conference, No.Th2bPo5.5, pp.1054-1059, Sept. 2016.
- [2] K. Greff, R. K. Srivastava, J. Koutn'k, B. R. Steunebrink and J. Schmidhuber: LSTM: A search space odyssey, arXiv:1503.04069v2, Oct. 2017.
- [3] F. A. Gers, J. Schmidhuber and F. Cummins: Learning to forget: Continual prediction with LSTM, Technical Report IDSIA-01-99, Jan. 1999.
- [4] S. Tokui, K. Oono, S. Hido and J. Clayton: Chainer: A next-generation open source framework for deep Learning, NIPS Workshop on Machine Learning Systems at Neural Information Processing Systems, Dec. 2015.