Paper:

A Comparative Study on Statistical Method and Neural Network in COVID-19 Forecasting

Naoki Dohi, Yukinobu Hoshino

185 Miyanokuchi, Tosayamada, Kami City, Kochi 782-8502, JAPAN
E-mail: 255055c@gs.kochi-tech.ac.jp
[Received 08/15/21; accepted 08/17/21]

Abstract. This study about forecast Japanese confirmed cases of the novel coronavirus to assist in decisions. There are statistical models and machine learning models for forecasting the time series data. Statistical models performed better than machine learning models. The experiment results were confirmed by both comparisons. Therefore, this paper would like to report the forecast confirmed cases of the novel coronavirus. SARIMA(Seasonal AutoRegressive Integrated Moving Average) and RNN (Recurrent neural network) are compared by RMSE (Root Mean Square Error). Results show that RNN(with vector inputs) was better than statistical models.

Keywords: Statistical Model, SARIMA, Machine Learning, Deep Learning, RNN, Time Series Forecasting, COVID-19, Novel Corona Virus

1. Introduction

1.1. Background

Currently, COVID-19 is a pandemic influenced over the world including Japan. There is a cumulative death toll is 1,500 as of July 6, 2021, in Japan. In addition, lifestyles have the strong require to change by the pandemic. The impact of the pandemic has also had a critical influence on economic activities. Japanese government requires refraining to whole business works, restaurants, industries at night time. Those requirements are called the COVID-19 Declaration of State of Emergency in Japan. In addition, the tourism business will also be economically affected because of self-imposed isolation. Therefore, it is crucial to forecast the number of people infected. The countermeasures need the number of people infected before the infection spreads to keep the economic activities. The history of infectious pandemics shows that we have a high probability the similar diseases in the future. Hence, it is essential to create a model as a study.

1.2. Purpose

According to the study by Spyros Makridakis et al. [1], it is recommended to compare statistical models and machine learning models while forecasting time series. Because almost studies have bias by using one side.

In this study, we compared statistical models and DeepLearning models. Statistical used Seasonal AutoRegressive Integrated Moving Average (SARIMA). DeepLearning used Recurrent Neural Network (RNN)[2]. RNN uses two types. one is RNN with multivariate input and single-variate output (Many to One). The other is RNN with multivariate input and output (Many to Many). Because it believes that Many to Many have a better prediction. In general, time series forecasting is often done with Many to One. However, we believe that Many to Many are better while considering the minimization of error with training data and label data. In addition, it is predicted dynamically by iterate predictions in test terms.

2. SARIMA

This section first discusses the autoregressive integrated moving average (ARIMA) to explain SARIMA. If it difine L^d as Eq. 1, d-order differencing series as $\Delta^d y_t$ (where $\Delta^0 y_t = y_t$) is formulated as Eq.(2). ARIMA is formulated as Eq.(3). Note that ϕ refers to the autoregressive coefficient (strength of correlation), ε_t refers to the error (following a normal distribution with mean 0 and variance σ^2), and θ refers to the moving average coefficient (strength of correlation).

$$L^d y_t = y_{t-d} \quad \ldots\ldots\ldots (1)$$

$$\Delta^d y_t = (1-L^d) y_t \quad \ldots\ldots (2)$$

$$(1-\sum_{i=1}^{p} \phi_i L^i)\Delta^d y_t = (1+\sum_{j=1}^{q} \theta_j L^j)\varepsilon_t \quad \ldots\ldots (3)$$

SARIMA applied to seasonal variation components such as one-week cycle and quarterly cycle with ARIMA. in this time, the model was applied to a one-week cycle. Fig.1 shows an overview of the SARIMA model. Note that y_t is the data at each time.

The data circled in blue are ARIMA terms. on the other hand, The data circled in red are SARIMA terms. From this, this can include long-term components in the model.

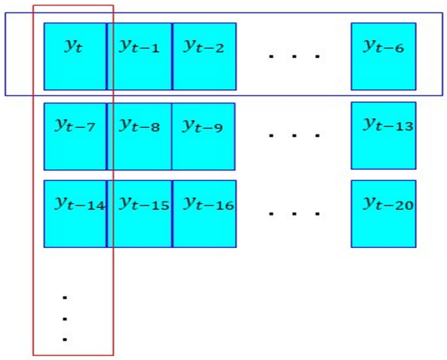

Fig. 1. SARIMA

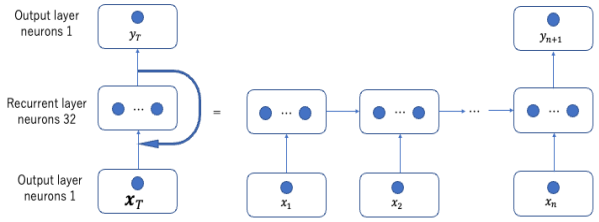

Fig. 2. Many to One

If the order of seasonal autoregressive, the order of seasonal moving average, the lag operator for the seasonal variation component and D-order seasonal difference is define as P, Q, L^{sD} and $\Delta_s^D y_t$ (but $\Delta_s^0 y_t = y_t$), SARIMA is formulated as in Eq. (4). Note that ϕ is the autoregressive coefficient (strength of correlation), θ is the moving average coefficient (strength of correlation), Φ is the seasonal autoregressive coefficient (strength of correlation), Θ is the seasonal moving average coefficient (strength of correlation), and ε_t is the error (following a normal distribution with mean 0 and variance σ^2).

$$(1-\sum_{i=1}^{p}\phi_i L^i)(1-\sum_{I=1}^{P}\Phi_I L^{sI})\Delta^d\Delta_s^D y_t = (1+\sum_{j=1}^{q}\theta_j L^j)(1+\sum_{J=1}^{Q}\Theta_J L^{sJ})\varepsilon_t \quad . . . . . (4)$$

3. RNN

RNN recursively inputs own outputs in hidden layers. The operations in the hidden layers is define as Eq.(5). Note that t is the time, $\boldsymbol{x}_t$ is the input, $\boldsymbol{h}_t$ is the hidden state vector (meaning past information), $\boldsymbol{W}_x$ is the weight for the input, and $\boldsymbol{W}_h$ is the weight for the hidden state vector.

$$\boldsymbol{h}_t = \tanh(\boldsymbol{x}_t\boldsymbol{W}_x + \boldsymbol{h}_{t-1}\boldsymbol{W}_h + \boldsymbol{b}) \quad . . . . . (5)$$

3.1. Relationship between inputs and outputs

RNN can have various relationships between input and output. In this study, we used two types. one is Many to One. the other is Many to Many. In this section, It is noted about those relationships

3.1.1. Many to One

Many to One can predicts eighth day by seven days inputs shown in Fig.2

3.1.2. Many to Many

Many to Many can predict the next seven days by seven days inputs shown in Fig.3.

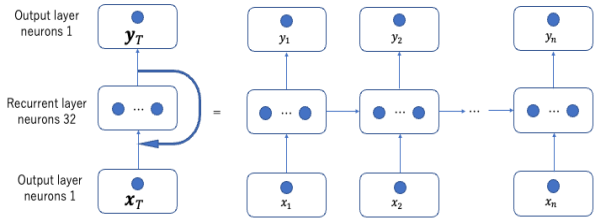

Fig. 3. Many to Many

In general, real time series data, such as stock prices and number of infections, behave chaotically. Therefore, the beginning error in the time series has a big influence on the prediction. Therefore, in RNN which is Many To One, if beginning error in the time series is big, the prediction error will be big. On the other hand, Many To Many can minimize the error with the labe data at each time.

4. Evaluation Function

The Root Mean Square Error (RMSE) is defined as Eq.(6). RMES was used to evaluate the prediction error.

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2} \quad . . . . . (6)$$

In Eq.(6), y_i means the observed value. $\hat{y}_i$ means the predicted value. n means the number of data. RMSE squares the error in order not to has negative value. Thus, RMES becomes a big value when the prediction is wrong.

5. Experiment

5.1. Experiment Order

This study was conducted using the following procedures.

1. Collecting data
2. Normalization of data and dataset creation for deep learning
3. Building four models
4. Model evaluation and comparison with evaluation functions

Firstly, the datasets were collected and processed.

In this study, use the dataset provided by Johns Hopkins University (JHU)[3]. This dataset includes the cumulative confirmed cases, cumulative recovered cases, and cumulative deaths for each country. Next, we created new dataset by differencing the dataset as $y_t - y_{t-1}$. This means that the new dataset includes newly infected cases, newly recovered cases, and new deaths in a day. The interval of the newly created dataset is from January 23, 2020, to October 28, 2020. For the training data, It used the data from January 23, 2020, to October 21, 2020. For the test data, It used the data from October 22, 2020, to October 28, 2020.

Secondly, two processing steps were applied to the dataset for deep learning. The first is normalization.

Normalization, which was conducted with Eq.(7) and Eq.(8).

$$x_{std} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad \ldots\ldots\ldots\ldots \quad (7)$$

$$x_{scaled} = x_{std}(max - min) + min \quad \ldots\ldots \quad (8)$$

In this case, x_{min} is the minimum value of the data, which is 0 people. x_{max} is the maximum value of the data, which is 3941 with vector input, which is 1762 with scalar input. Note that the range of normalization was between -1 and 1. It means max is 1, min is -1.

The second is the creation of the training and label data.

We created the dataset as shown in Fig.4 and Fig.5.

Fig.4 is the dataset for Many to One, where we created the dataset with a week as training data and eighth day as the label data.

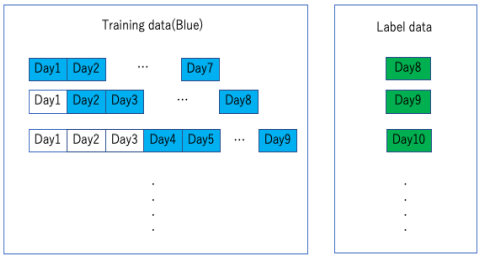

Fig. 4. Dataset for Many to One

Fig.5 shows the dataset for Many to Many, where one week as training data, the next week as label data.

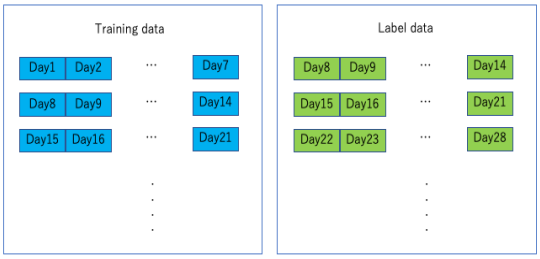

Fig. 5. Dataset for Many to Mamy

Thirdly, it built four models.

The first is SARIMA. Grid search was conducted with parameters p,d,q,P,D,Q in the range of $p = 1$ to $p = 3$, $d = 0$ to $d = 1$, $q = 0$ to $q = 3$, $P = 0$ to $P = 3$, $D = 0$ to $D = 2$, and $Q = 0$ to $Q = 3$ for building SARIMA. As a result, we builded a SARIMA$(3,1,2)(0,2,3)_7$ model with small prediction error.

SARIMA was fitted with new infections without normalization.

The second is RNN which is Many to One.

RNN which is Many to One has output the number of the new infections and inputs the normalized number of new infections as input. The number of neurons in the input and output layers was set to one, and the number of neurons in the hidden layers was set to 32. Adam was used as the optimization algorithm.

The third is RNN which is Many to Many with scalar (number of new infected) inputs.

RNN which is Many to One with scalar (number of new infections) inputs has outputs the number of new infections and inputs the normalized number of new infections. The number of neurons in the input and output layers was set to one, and the number of neurons in the hidden layers was set to 32. Adam was used as the optimization algo-

rithm.

The fourth is RNN which is Many to Many with vector input (number of new infections, new recovered people, and new deaths). In RNN which is Many to Many with vector input, it inputs the number of new infections, new recovered people, and new deaths, and outputs the number of new infections. The number of neurons in the input layers was set to three, in the output layers was set to one, and in the hidden layers was set to 32. Adam was used as the optimization algorithm.

Finally, each model is evaluated by RMSE values.

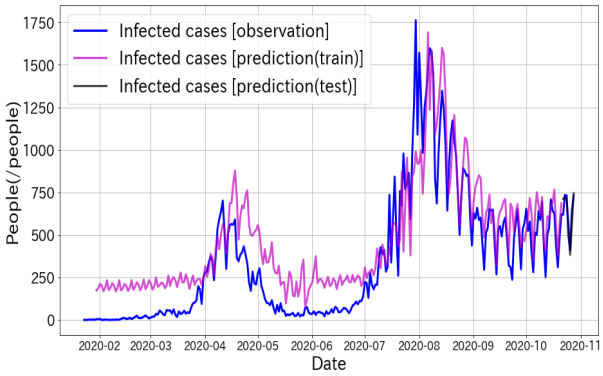

Fig. 6. Result of Many to Many with vector inputs

5.2. Results

The results from the four models were compared with RMSE values. RMSE of SARIMA was 70.19. RMSE of RNN which is Many to One was 109.57. RMSE of RNN which is Many to Many with scalar input was 110.39. RMSE of RNN which is Many to Many with vector input was 59.04.

According to these RMSE values, the smallest RMSE value in the test terms was RNN of Many to Many with inputs of the number of new infected, new recovered, and new death.

In addition, while comparing statistical model and deep learning model, RMSE value of the Many to Many RNN with vector inputs was the smallest, indicating that the deep learning model was able to make better predictions. This result is different from the result described in the study by Spyros Makridakis et al[1].

6. Considerations

In this study, RMSE of RNN which is Many to Many with vector inputs was the smallest. This can be considered as a result of the vector inputs and as no result Many to Many. Because there is no big difference in RMSE between RNN which is Many to One and RNN which is Many to Many of scalar input, Therefore, this difference is caused by whether the input is vector or not.

Thus, in future research, further improvement in accuracy can be expected by increasing variables. However, since multicollinearity may occur, it is necessary to calculate the correlation between time series data.

Also, in this study, The bias loading was observed prediction in training term using RNN which is Many to Many with vector input shows as Fig.6.

This is expected to be improved by normalization for each variable.Because, the terms observed is terms which is increased recovered people, and present variables is normalized by maximum value of all variables.Therefore, in future studies, we thought that model can learning more appropriately by normalization with the maximum value of each variable.

7. End

In the results of this study, deep learning gave better results than statistical models. We believe that this is due to the vector inputs. We also believe that the bias load when predicting the training term is occurred by normalization range.

Therefore, we will continue the verification as we believe that better models can be built by increasing variables and proper normalization.

Acknowledgements

I would like to express my heartfelt gratitude to Associate Professor Hoshino and the members of our laboratory for their guidance in writing this paper.

References:

[1] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. "Statistical and machine learning forecasting methods: Concerns and ways forward," PloS one, Vol.13, No.3, pp.e0194889, 2018.

[2] Jeffrey L Elman. "Finding structure in time," Cognitive science, Vol.14, No.2, pp.179–211, 1990.

[3] https://github.com/CSSEGISandData/COVID-19