Implementation of SimpleRNN and LSTMs based prediction model for coronavirus disease (Covid-19)

Priyanka¹, A Kumari² and M Sood³
¹,²,³Himachal Pradesh University, Shimla, India

E-mail: ¹roypriyanka970@gmail.com, ²ankkitaa211996@gmail.com, ³soodm_67@yahoo.com

Abstract. Deep learning is a powerful technique which is inspired by the structure as well as processing power of the human brain. This technique uses deep neural network to perform complex tasks such as time series prediction, image classification, and cancer detection. In this research work, we used Covid-19 time series datasets and with the help of deep learning we built the model for prediction of Covid-19 cases. For the model building, we used two deep learning neural networks, Recurrent Neural Networks (RNN) and Long Short Term Memory Networks (LSTMs). We built a prediction model using RNN in the first instance and subsequently the second model was built using LSTMs. Out of these two neural networks, we got promising results from the model based on LSTMs with an overall accuracy of 98%. As the cases of Covid-19 are increasing day-by-day at a very high rate, we proposed these models using neural networks to help in predicting the future trends of Covid-19 confirmed, deaths and recovered cases.

Keywords. Deep Learning, Covid-19, Recurrent Neural Network, Long Short Term Memory Networks, Dataset, REctified Linear Unit

1. Introduction
In the present scenario, the world is going through tough times as a novel form of a biological virus known as coronavirus has emerged that has devastated many lives along with the economies of most of the countries. Coronaviruses are named so because in the electron microscope, they create an image of “solar corona”. Earlier the coronaviruses have caused outbreaks of Severe Acute Respiratory Syndrome CoronaVirus (SARS-CoV) in China and the Middle East (Middle East Respiratory Syndrome CoronaVirus (MERS-CoV)) [1]. The Coronavirus disease 2019 (Covid-19) is a contagious disease caused by Severe Acute Respiratory Syndrome CoronaVirus 2 (SARS-CoV-2) [2]. This virus was first identified in December 2019 in China and after that it spread all over the world. It is a form of zoonotic disease and spreads by coming in close contact with infected people. The organs, which are mostly affected by this virus are the lungs and the respiratory system of the human body.

This deadly virus has already killed more than half a million people around the globe in the absence of any credible cure till date. Due to incomplete understanding of the ways it spreads and affects the human body, the number of people being affected as well as killed globally by this disease is increasing at a very high pace day-by-day. The intensity of this deadly disease can be gauged from the fact that the World Health Organization (WHO) had declared it as a pandemic in the initial phases of...
its spread only. The experts from various fields are already working intensely on how to eradicate this menace and data scientists are also among these experts. One of the ways the Data Science experts contribute to this cause is through building prediction models for the future behavior of the disease so that it can be predicted with accuracy up to what extend this virus can go further. And, as such, Deep Learning (DL) can provide adequate techniques for building such kind of prediction models.

The concept of DL was first introduced by Rina Dechter as an extension to ML [3] as concurred in the literature while this concept was introduced to ANN by Aizenberg in 2000 [4]. Deep learning (DL) is a subset of Machine Learning (ML) which in turn is a subset of Artificial Intelligence (AI). AI enables a machine to mimic human behavior while ML supports techniques to make AI achieve it through algorithms trained with data and DL is a type of ML inspired by the structure of the human brain. In terms of DL, this structure is termed as an Artificial Neural Network (ANN). In ML, when we train the machine we need to specify the features to the machine on the basis of which a machine is supposed to perform a task. These features are extracted by the neural networks without any human intervention and such type of independence comes up at a cost of having much higher volumes of data to train the machine.

A neural network supposedly has three layers named as the input layer, hidden layer and output layer and neurons are the core entity of this neural network. The processing of information takes place in these neurons which constitute different layers of the network. The information in the network is transmitted from one layer to another through connecting channels. Since each channel has a value attached to it, these are called as weighted channels. All neurons in the neural network have a unique number associated with it called as bias. This bias is added to the weighted sum of inputs reaching the neurons and applied to a function known as the activation function. The result of the activation function determines if a neuron gets activated. Every activated neuron passes on information to the succeeding layers and this continues up to the second last layer. The one neuron activated in the output layer corresponds to the output digit. The weights and bias are continuously adjusted to produce a well-trained neural network.

There are different types of neural networks but we have used Recurrent Neural Networks (RNN) and Long Short-Term Memory Networks (LSTMs) in this research work and the reason behind choosing these networks is that both of these networks are specifically capable of analyzing the time series data to predict the future trends. In this research work we have used COVID19 [5] time series data for analyzing and predicting the Covid-19 cases.

Figure 1. Structure of RNN [6], in this x is the input layers, h is the hidden layers (with loop back for accessing the previous output of each time-steps) and y is the output layers.

RNN is based on David Rumelhart’s work in 1986 and John Hopfield in 1982 [7]. RNN are used to process time series data and sequential data [8]. These type of networks are also called advanced feed forward neural networks because in simple feed forward neural network information always move only in one direction, i.e. only goes from input layer to hidden layer to output layer, it never goes back. But in RNN it has connections pointing backward too meaning thereby that the data can move in both forward and backward directions [9]. RNN also shares the parameters across different time-steps. It consists loops in the layers which means that a neuron receiving inputs and producing an output can be...
allowed to send that output back to itself too. So, in this way, the RNNs use two types of inputs, i.e. present input denoted by \( x \), and the other input from the output of the previous time-steps denoted by \( y_{t-1} \) [10]. The Structure of RNN is depicted in Figure 1. As can be seen in this figure, there are three levels of layers in this diagram, input layer represented by \( x \), output layer represented by \( y \) and the hidden layer is represented by \( h \).

RNN leads the vanishing gradient point error so, to deal with this problem a new model was introduced named as Long Short-Term Memory Networks (LSTMs) also called Advanced RNN. LSTMs are also denoted by LSTMs-RNN. LSTMs introduced by Horchreiter and Schmidhuber in 1997 are the special kinds of neural networks that are capable of remembering information for longer periods of times. These LSTMs have a chain like structure having different repeating modules. With the help of LSTMs-RNN, the RNN learning process really becomes better [11] and supportive to avoid the vanishing gradient and explosion of gradient errors. This means that this special category of RNN can take long term dependencies and long length sequenced data [12]. We have used Rectified Linear Unit (ReLU) activation function in LSTMs; and tanh and sigmoid activation functions in simple RNN. We applied these neural networks on the available dataset containing data corresponding to the Covid-19 scenario across the globe and measured six (6) keras metrics to predict corona virus cases. Corona virus is harming people health and leads to death. Since a large majority of people in various countries of the world are getting affected by this disease and with no cure till date available around the corner, there is an urgent need to predict the future cases from the existing data. This paper is an attempt to build prediction models based on Deep Learning through which we can achieve best performance results for predicting three parameters for future namely, confirmed corona virus Covid-19 positive cases, recovered from this disease cases and the death from this disease cases.

The novelty of this research work is presented below:

1. All the three datasets of covid-19 [5] have been considered for the purpose of training and testing of predictive models and merging process has culminated into 2 datasets in the ratio of 100:100 from these three datasets.
2. Two types of DL based neural networks have used, RNN and LSTMs for predicting confirmed, recovered and death cases of corona cases and we have also measured and compared the performances of prediction models based on both of these neural networks.
3. Three types of activation functions have been used to calculate the output of the RNN and LSTMs networks named as RELU, tanh and sigmoid.
4. We have listed the limitations of simple RNN encountered during our work.
5. The results have shown that the performance of LSTM is better for predicting the corona virus cases.

1.1. Objectives

The objectives of this research work are listed below:

1. To compare and analyze the performances of prediction models based on two neural networks, RNN and LSTM-RNN to find out which one performs better in predicting confirmed positive cases, recovered cases and mortality.
2. To achieve best predicted output using an appropriate activation function.
3. To predict the future cases of corona virus on the bases of existing datasets.

1.2. Paper structure

This paper is divided into four sections in which section 2 explains the research methodology that was followed for achieving the objectives of this paper, section 3 presents the results of our experiments and their analysis whereas section 4 concludes the paper with the direction for the future work.

2. Research Methodology

The research methodology is the stepping stone to achieving valuable outcomes. Figure 2 depicts the methodology that we have followed so as to achieve the objectives of this research work. After the
basic preprocessing of the data in the specified datasets, two proposed prediction models based on a) RNN and b) LSTM networks have been used on these datasets where the ratio of training data to testing data was kept at 70:30. Three layers used for first network model were SimpleRNN, Dropout and Dense along with two activation functions, Tanh and Sigmoid. Whereas for the later model, three layers used were LSTM, Dropout and Dense along with the ReLU activation function. The six metrics used for evaluating the performances of both of these models are: accuracy, MAE, MSE, logcosh, MSLE and cosine similarity.

2.1. Dataset
In this section, a brief introduction to the dataset is being provided. The datasets used for conducting this research work are shown in Table 1.

![Diagram](https://via.placeholder.com/150)

**Figure 2.** Research methodology used in our research work.

This table shows the three covid-19 datasets that we have used in our research work i.e. covid_19_confirmed, covid_19_deaths and covid_19_recovered. All these three datasets have been acquired from Kaggle [5]. These datasets contain the time series data about the people infected by Covid-19 in all the countries worldwide. This time series data taken in these datasets for our experiments is from 22nd January 2002 till 30th June 2020.

| S. No. | Dataset        | No. of rows | No. of columns |
|--------|----------------|-------------|----------------|
| 1      | covid_19_confirmed | 266         | 165            |
| 2      | covid_19_deaths   | 266         | 165            |
| 3      | covid_19_recovered | 253         | 165            |

2.2. Experimental Setup
In this section, we have discussed about the experimental setup used by us for conducting our research work. For this, firstly we have installed anaconda as it is a free and open-source [13]. In anaconda
navigator, we have used jupyter notebook and in jupyter notebook we have further installed tensorflow. As tensorflow is an open-source library created for tasks with heavy numerical computation with its faster compiler. Tensorflow have an extensive built in support for deep learning and neural network. So we have used the tensorflow and after that we have also used keras which is tensorflow API written in python. All the experimentation in this research work is conducted using the python language.

2.2.1. Data Pre-processing. The pre-processing step basically includes the scaling, normalization, transformation and cleaning of data. The Table 2 below shows the detail of the dataset after pre-processing. In this, firstly we have merged those rows which contain the data of various states of a country. So after merging all such rows, we are left with 188 rows in each dataset. Next we have removed Lat, Long and Province/State features as they are not much helpful for achieving the target and left with 162 columns in all the three datasets.

| S. No. | Dataset            | No. of rows | No. of columns |
|--------|--------------------|-------------|----------------|
| 1      | covid_19_confirmed | 188         | 162            |
| 2      | covid_19_deaths    | 188         | 162            |
| 3      | covid_19Recovered  | 188         | 162            |

This table shows the number of rows and columns left in each dataset after pre-processing the data. In all these three datasets, we have one feature named Country so for the transformation of this feature we have used label encoder as it converts categorical features into numeric values. After pre-processing these datasets, we have biased the covid_19_confirmed dataset with covid_19_deaths and covid_19Recovered datasets respectively and got two biased datasets i.e. confirmed & deaths and confirmed & recovered with 376 rows in each.

2.2.2. Train Set and Test Set. In this research work, train set is used to build the training model and test set is used to test the training model. Here we have used the ratio of 70:30 for partitioning the Covid-19 biased datasets into train and test which means that we have trained our proposed models with 70% of data and tested this trained models with rest of the 30% of respective datasets so as to evaluate the performances of the proposed models.

2.2.3. Recurrent Neural Network (RNN). In this research work, we have to use RNN because we have chosen the time series data from January to June this year related to Covid-19 situation throughout the world. RNN is supported by the characteristic property of using time series data for making predictions of the future. So, for implementing Simple RNN, we selected keras model = Sequential () to process sequences of integers embedding each integer into a 64-dimensional vector. Later, we used the following three layers of keras. SimpleRNN (to provide the number of units, input shape and input features) layer is fully connected RNN where the output form previous timestep is to be fed to the next timestep. Then next layer used is the dropout layer which can tackle overfitting in the RNN based model. The maximum dropout used on our model is 0.5. After that we have used dense layer to receive the input from the all neuron so that it can be passed on to the compiler.

The two activation functions used are: a) tanh activation function, a zero centered hyperbolic tangent activation function that combine input signals and generates the outputs in the ranges from -1 to +1 (that’s why it is called zero centered function) i.e. tanh= -1 to +1. b) sigmoid activation function, a logistic function with ranges from 0 to 1, is a non-zero centered function to combine the inputs based on intensity and similarity and once inputs are combined then it generates the output. We have used six keras metrics to evaluate the performance of simpleRNN model. With the use of tanh and the
sigmoid activation function in simpleRNN, we have calculated keras six metrics in order to evaluate performance of our simple RNN model. The performance of prediction model based on RNN was not as per expectations as there was no stability in outcomes. Every time we applied our RNN based model to the datasets, different outcomes were being generated and the results displayed leading vanishing gradient point error. Therefore, we decided to build another model based on advanced RNN also called LSTM.

2.2.4. Long Short Term Memory Networks (LSTM). In this research work, after getting poor outcomes from the model based on simpleRNN, we have used LSTM based prediction model to predict Covid-19 cases. LSTMs have cell states in such ways that these can selectively remember or forget things. Three gates work for the cell state. 1) Forget gate: Forget gate removes that information from the cell states which is no longer needed. 2) Input gate: This gate is responsible for the addition of information to the cell state. 3) Output gate: This gate is responsible for selecting useful information and generating the output [14]. So with the help of these three gates we have created three layers for building this LSTMs model: LSTMs layer which provide the number of units, input shape and input features, dropout layer which tackles overfitting in the network and dense layer that receives the input from all the neurons and passes on to the compiler. In LSTMs, we have used the same layers as those of the simpleRNN. We have carried out two different steps; firstly we have set fixed random seed for reproducibility. Then we have used ReLU activation function. For negative (-ve) values, it gives outcome in the form of a ‘0’ and for positive (+ve) values, it is linear in nature. So,

\[ f(x) = \{x, \text{if } x \geq 0, 0 \text{ if } x < 0 \} \] where x is the input values and this ReLU also support nonlinear dependencies.

Using this ReLU in LSTMs we have again calculated the six keras metrics to explore the performance of the proposed model including the vanishing gradient point error for the prediction of corona virus cases.

2.2.5. Keras metrics. For evaluating the results of our experimentation, we have used six keras metrics, i.e. Accuracy, Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Squared Logithmic Error (MSLE), cosine similarity, and log cosh error (logcosh). Out of these six metrics, accuracy is an accuracy class metric whereas the rest of the five are regression metrics. The accuracy is used to calculate how often a prediction equals to the label. The MSE metric is used to compute the mean squared error between \( y_{true} \) and \( y_{pred} \) whereas the MAE metric is used to compute the mean absolute error between labels and predictions. The MSLE metric is used to compute the mean squared logarithmic error between \( y_{true} \) and \( y_{pred} \) and the cosine similarity metric is used to compute the cosine similarity between labels and predictions. The logcosh metric stands for the logarithm of hyperbolic cosine error. This is used to compute the hyperbolic logcosh of the prediction error [15]. The best value for the metrics, accuracy and cosine similarity is supposed to be nearing 1 and for other four metrics, it should be nearing 0.

3. Results & Analysis

This section presents the results obtained through the experiments performed using our proposed prediction models and their analysis. We have used two neural networks i.e. RNN and LSTM in this study so as to predict three parameters related to Covid-19 situation throughout the world: a) confirmed cases, b) recovered cases, and c) death cases. The performances of these neural networks were evaluated using six keras metrics as stated in the preceding section.

3.1. Results of RNN
In this research work, the first neural network used was RNN as we were working on time series Covid-19 dataset. RNN is used to predict the future using existing time series data. So, for
implementing RNN we used keras model = Sequential () to process sequence of integers and embeds each integer into a 64-dimensional vector. Then we have used three layers of keras namely SimpleRNN, dropout layer and dense layer. The SimpleRNN layer is used to provide the number of units, input shape and input features. This layer is fully connected where the output form previous timestep is to be fed to next timestep. The dropout layer is used to tackle overfitting in our RNN model and we have used maximum dropout as 0.5 in our model. The dense layer is used to receive the input from the all neuron and then pass those inputs to the compiler. After this, we have used two activation functions and those were tanh activation function and sigmoid activation function. With the use of tanh and sigmoid activation function in SimpleRNN, we calculated six keras metrics in order to the evaluate performance of our model. But the performance of RNN we have obtained was worst because there was no stability in the outcomes. Whenever we ran our RNN model on the datasets multiple times, different values were obtained every time for the almost all the metrics as shown in Table 3 for three separate executions of the proposed model. Also, we witnessed the vanishing gradient point error too. Therefore, we were left with no option but to discard this model.

### Table 3. Varying values of keras metrics for three executions of SimpleRNN prediction model.

| Dataset          | Run | Accuracy | Cosine Similarity | MSE   | MAE   | MSLE   | Log Cosh Error |
|------------------|-----|----------|-------------------|-------|-------|--------|----------------|
| Corona Confirmed | 1st | 0.9085   | 0.6178            | 0.0043| 0.02  | 0.0021 | 0.0019         |
|                  | 2nd | 0.8292   | 0.5764            | 0.0043| 0.02  | 0.0022 | 0.0019         |
|                  | 3rd | 0.8841   | 0.6363            | 0.0042| 0.0206| 0.0021 | 0.0019         |
| Confirmed & Deaths | 1st | 0.8835   | 0.6401            | 0.00424| 0.02526| 0.002 | 0.0019         |
|                  | 2nd | 0.875    | 0.6242            | 0.0045| 0.0254| 0.0021 | 0.0002         |
|                  | 3rd | 0.8437   | 0.6841            | 0.0038| 0.0248| 0.00185| 0.0017        |
| Confirmed & Recovered | 1st | 0.8352   | 0.606             | 0.00474| 0.0266| 0.00236| 0.0021        |
|                  | 2nd | 0.8835   | 0.6298            | 0.0048| 0.0272| 0.00236| 0.0022        |
|                  | 3rd | 0.6477   | 0.57              | 0.005 | 0.027 | 0.00248| 0.0023        |

### 3.2. Results of LSTMs
After discarding the prediction model based on SimpleRNN due to its poor performance, we used a prediction model based on LSTM to predict the Covid-19 cases from the datasets. LSTMs have cell states and can selectively remember or forget information. The three gates that worked for the cell state were forget gate, input gate and output gate. So with the help of these gates we have used three layers namely LSTM layer, Dropout layer and Dense layer for building the LSTMs model. In this, we executed two different steps from that of SimpleRNN. Firstly, we set fixed random seed for reproducibility and then used Rectified Linear activation function (ReLU). With this ReLU, we calculated the six keras metrics for LSTMs and obtained best results. Moreover, the vanishing gradient point error was also reduced. The Table 4 below shows the results of LSTMs based prediction model and Fig. 1 gives its graphical representation.

### Table 4. Results of keras metrics for LSTMs based prediction model.

| Dataset          | Accuracy | Cosine Similarity | MSE   | MAE   | MSLE   | Log Cosh Error |
|------------------|----------|-------------------|-------|-------|--------|----------------|
| Corona Confirmed | 0.9756   | 0.9198            | 0.0022| 0.013 | 0.0011 | 0.001         |
| Confirmed & Deaths | 0.9886   | 0.9644            | 0.0018| 0.0105| 0.0009 | 0.0008        |
| Confirmed & Recovered | 0.9857   | 0.9437            | 0.0021| 0.0141| 0.0011 | 0.001        |
As we can see in this table, the values of accuracy and cosine similarity are high nearing almost ‘1’ while the values of all the other errors and loss are extremely low almost nearing ‘0’. So we can say that our prediction model is successfully trained and tested because in keras metrics the value of the errors should be low while the values of accuracy and cosine similarity should be high.

Figure 3. Performance of LSTM based prediction model through Keras Metrics on three datasets.

The Fig. 3 shows that our proposed prediction model can predict the ‘Corona confirmed’, ‘confirmed & deaths’ and ‘confirmed & recovered’ cases with 98%, 99% and 99% accuracy respectively. The values of cosine similarity for these three cases are 0.92, 0.96 and 0.94 respectively. The values of Mean Squared Error for these three cases are 0.0022, 0.0018 and 0.0021 respectively whereas values of Mean Absolute Error are 0.013, 0.011 and 0.014 respectively, values of Mean Squared Logarithmic Error are 0.0011, 0.0009 and 0.0011 respectively and that of LogCosh Error are 0.001, 0.0008 and 0.001 respectively. These values obtained from our prediction model after its training and testing on the given datasets are very close to the perfect values of these metrics. From these results, we conclude that LSTM has given the stable prediction results with a very high accuracy for the time series Covid-19 datasets while with SimpleRNN we were able to obtain varying results only. Based on these observations, we can conclude that our prediction models based on LSTM neural network can predict the ‘Corona confirmed’, ‘confirmed & deaths’ and ‘confirmed & recovered’ cases with an overall accuracy of around 98% validated with bare minimum possible error values.

4. Conclusion
The Covid-19 pandemic has been playing havoc with the lives of human beings in almost every nook and corner of most of the countries on this earth. The situation is going from worse to worst day by day. With no cure available for it till date coupled with the fact of having very bleak chances of its complete eradication in near future, it was considered as a good idea to use deep learning models to make future predictions in this regard. In this research work, using time series datasets of all the countries affected by Covid-19, we have proposed two deep learning based prediction models. The selected three datasets in the form of time series data were subjected for experimentation to two prediction models based on neural networks RNN and LSTMs. Python language has been used in the development and execution of these neural networks and the results were evaluated using six keras metrics. Firstly, the SimpleRNN based prediction model was used on these datasets but the results in
the form of values of six keras metrics kept on varying for every such execution leading to unstable outcomes. Every time this neural network was run, different results were achieved further leading to vanishing gradient point error. Later, LSTM neural network was used for building the second prediction model. The results obtained for all the six keras metrics after executing this model were not only quite promising but were also stable and did not vary even after a number of its executions on the datasets under consideration. An overall accuracy of 98% has been achieved by this proposed prediction model based on LSTM for ‘Corona confirmed’, ‘confirmed & deaths’ and ‘confirmed & recovered’ cases along with the reduced value of vanishing gradient point error. Currently, the authors are working on these predictions for individual countries and in future, we will try to compare these deep learning based prediction models for Covid-19 with those based on machine learning.

References
[1] Murray PR, Rosenthal KS and Pfaller MA 2020 Medical Microbiology Elsevier Health Sciences
[2] https://en.m.wikipedia.org/wiki/Coronavirus_disease_2019 last accessed on 7/22/2020 at 22:25
[3] Dechter R 1986 Learning While Searching Constraint-Satisfaction Problems Fifth National Conference on Artificial Intelligence pp 178-183
[4] Aizenberg I, Aizenberg NN and Vandewalle JPL 2000 Multi-Valued and Universal Binary Neurons: Theory, Learning and Applications Springer Science & Business Media
[5] https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset/data# last accessed on 7/15/2020 at 23:24
[6] https://www.researchgate.net/figure/Recurrent-neural-networks-structure_fig4_326884617 last accessed on 7/1/2020 at 1:26 pm
[7] https://en.wikipedia.org/wiki/Recurrent_neural_network#:~:text=Hopfield%20networks%20%20%20%20%20%20%20,an%20RNN%20unfolded%20in%20time last accessed on 7/22/2020 at 12:26 pm
[8] https://www.sciencedirect.com/topics/engineering/recurrent-neural-network last accessed on 7/21/2020 at 4:26 pm
[9] Poznyak TI, Chairez Oria I and Poznyak AS 2019 Background on Dynamic Neural Networks Ozonation and Biodegradation in Environmental Engineering Elsevier Amsterdam The Netherlands pp 57-74
[10] Gupta TK and Raza K 2019 Optimization of ANN architecture: a review on nature-inspired techniques Machine learning in bio-signal analysis and diagnostic imaging pp 159-182
[11] Kumar J, Goomer R and Singh AK 2018 Long short term memory recurrent neural network (lstm-rnn) based workload forecasting model for cloud datacenters Procedia Computer Science pp 676-682
[12] Ye Q, Yang X, Chen C and Wang J 2019 River Water Quality Parameters Prediction Method Based on LSTM-RNN Model Chinese Control And Decision Conference (CCDC) 2019 pp 3024-3028
[13] https://en.wikipedia.org/wiki/Anaconda_(Python_distribution) last accessed on 20-07-2020 at 13:04
[14] https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/ last accessed on 7/22/2020 at 4:06 pm
[15] https://keras.io/api/metrics/ last accessed on 5/7/2020 at 1:31 am