Predicting confirmed cases and fatalities globally due to COVID-19 using LSTM

Syed Kashif Mujtaba Kamoonpuri

Broad College of Business, Michigan State University, East Lansing, MI, USA

kamoonpu@msu.edu

***Abstract*–On March 11, 2020, the World Health Organization (WHO) declared COVID-19 as a global pandemic. As of today, mid August 2020, there have been nearly 750,000 deaths with over 20.8 million cases of COVID-19. The objective of this paper is to build stacked Long Short Term Memory (LSTM) neural network model to predict the number of confirmed cases and fatalities occurring due to COVID-19. The model was trained and tested on different hyperparameters. The obtained models were evaluated on the basis of Mean Squared Error (Loss) and Root Mean Squared Error (RMSE).**

***Keywords*–COVID-19, LSTM, Neural Network, RNN**

1. INTRODUCTION

The outbreak of coronavirus disease 2019 (COVID-19) has created a global health crisis that has had a major impact on the world and our everyday lives. It is estimated that the pandemic will decrease the global economy by 3% this year. While a vaccine for this disease is still in production, it has been scientifically proven that wearing a mask, maintaining social distance and regularly washing hands prevents the spread of coronavirus. Many government agencies are relying on analytical tools and machine learning algorithms that predict the number of confirmed cases and deaths to make data driven decisions and policies. In this paper, we have used time-series data of reported COVID-19 cases and deaths around the world to train and test a recurrent neural network (LSTM) model. Unlike feedforward neural networks, recurrent neural networks (RNNs) have feedback connections, which allows it to process sequences of data. However, RNNs face the issue of vanishing gradient when performing backpropagation, which leads to very small gradients in some cases. A standard RNN structure has the following output:

$$h_t = \tanh(W(h_{t-1}x_t))$$

In this above equation, W is the weight of the layer, h_{t-1} is the output from the previous layer, h_t is the output from layer at time-state t, x_t is the input and $tanh$ is the activation function.

LSTMs are a special kind of RNN algorithms that are capable of learning long-term dependencies. LSTMs have 'tanh' activation function, designed to overcome the vanishing gradient problem.[1] Figure 1 shows the architecture of a basic LSTM model.

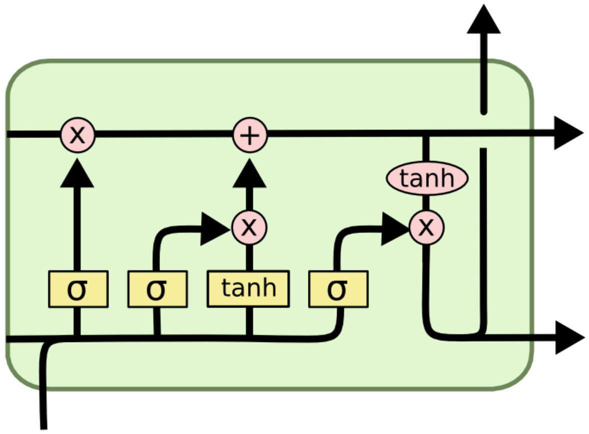

Fig. 1: Schematics of LSTM structure

2. COVID-19 DATASET

The data used for this paper was downloaded from the GitHub repository of Center for Systems Science and Engineering, Johns Hopkins Whiting School of Engineering.[2] The dataset contains time-series data of confirmed cases and fatalities due to COVID-19 for 188 countries, from January 22, 2020 to August 10, 2020. The dataset was originally in wide format and was converted to long format for modelling purpose. Figure 2 shows the total number of confirmed cases and fatalities with time, globally.

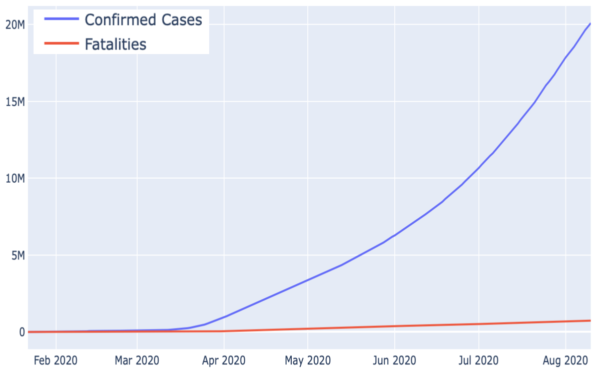

Fig. 2: Total number of confirmed cases and fatalities due to COVID-19

Figure 3 shows the number of confirmed cases and fatalities for Brazil, India, California (US), Michigan (US), New York (US) and United Kingdom with time.

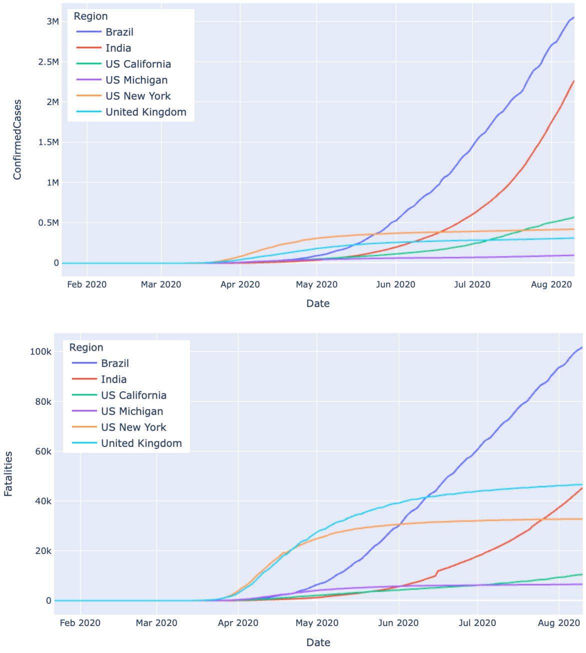

Fig. 3: (a) Number of confirmed cases for each region with time, (b) Number of fatalities for each region with time

The values for confirmed cases and fatalities was scaled using MinMaxScaler(). Finally, ten day lag was added for both confirmed cases and fatalities for each row, before the dataset was split into train and test set.

3. CREATING BASE MODEL

To predict the number of confirmed cases and fatalities due to COVID-19, I used 3-layer neural network with stacked LSTMs. The neural network was created using Keras on top of TensorFlow framework and has more than 50,000 trainable parameters. The input shape for the model is (10,1), which contains 10-day lag for confirmed cases or fatalities. The architecture for the neural network model is shown in figure 4.

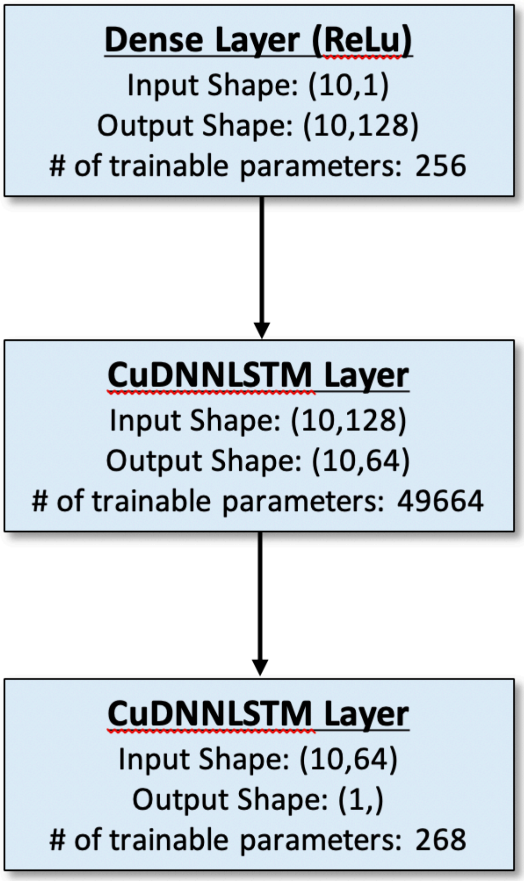

Fig. 4: Neural Network Architecture

The base neural network model was optimized using Stochastic Gradient Descent (SGD) with a learning rate of 0.01. Stochastic Gradient Descent is an extension of Gradient Descent algorithm used for backpropagation. Gradient descent is an iterative optimization algorithm that finds the local minima for the cost/loss function by analyzing the entire dataset and hence is computationally expensive. On the other hand, Stochastic Gradient Descent randomly picks only one data point in each iteration to perform backpropagation, thus reducing the computational time and cost.

Initially, the model was run for 10 epochs, but it was observed that the model was under-fitting (as shown in figure 5.

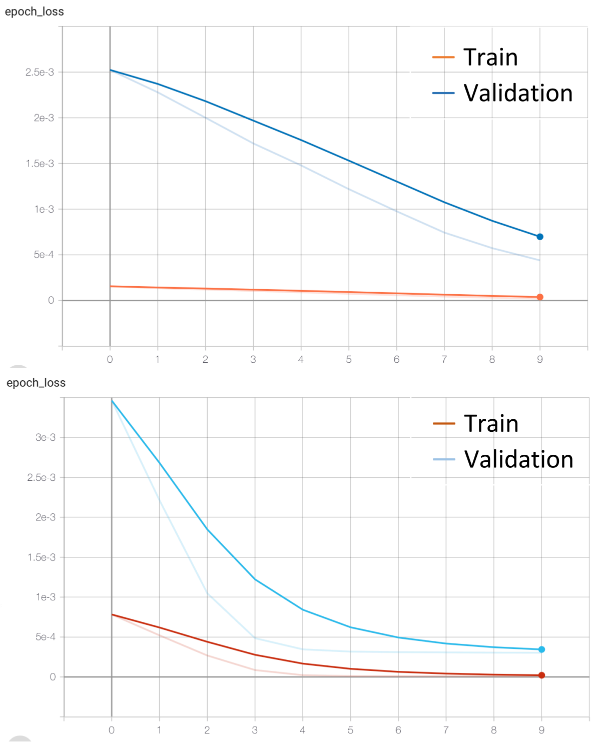

Fig. 5: (a) Epochs vs loss for predicting confirmed cases (10 epochs), (b) Epochs vs loss for predicting fatalities (10 epochs)

As observed from figure 5, we can see than the validation loss was still decreasing and had not reached the saturation value. The model was evaluated using Mean Squared Error (Loss) and RMSE (Root Mean Squared Error). The results of the model are as follows:

Parameters	Target Variable	Validation Loss	Validation RMSE
SGD 10 epochs	Cases	0.0007	0.027
	Fatalities	0.0003	0.019

Table 1: Model Performance (Epochs=10)

To overcome the under-fitting issue, the model was trained with 50 epochs, without changing any other hyper-parameters. Figure 6 shows the epochs vs loss graph for this graph.

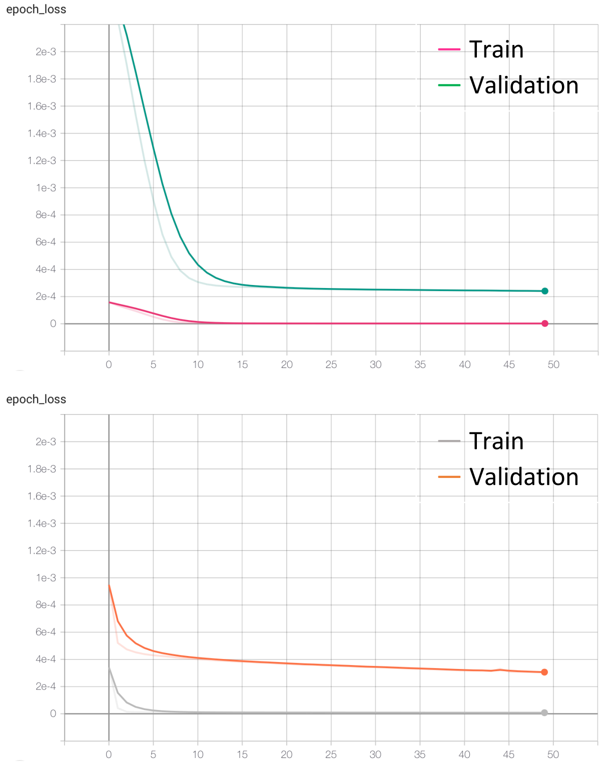

Fig. 6: (a) Epochs vs loss for predicting confirmed cases (50 epochs), (b) Epochs vs loss for predicting fatalities (50 epochs)

As observed from figure 6, train and validation loss reached saturation level at 50 epochs for both confirmed cases and fatalities.

The final Mean Squared Error (Loss) and RMSE (Root Mean Squared Error) of the model are as follows:

Parameters	Target Variable	Validation Loss	Validation RMSE
SGD 50 epochs	Cases	0.0002	0.015
	Fatalities	0.0003	0.017

Table 2: Model Performance (Epochs=50)

After comparing the performance of both models, we can conclude that the model with 50 epochs performs much better than model with 10 epochs.

4. VISUALIZATION

In order to evaluate model performance, I have predicted and visualized the performance for the following regions (displayed from left to right in figure 7 & 8):

1. Ontario, CA
2. Arizona, US
3. California, US
4. Florida, US
5. Georgia, US
6. Louisiana, US
7. Michigan, US
8. Texas, US

These cities have been strategically chosen, so that we have a mix in terms of cities experiencing second COVID-19 wave. Arizona, California, Florida and Texas are all currently going through the second wave of COVID-19.

After predicting the number of confirmed cases and fatalities for each of these cities, I inverse transformed the normalized predicted values.

The actual and predicted values of confirmed cases and fatalities for the aforementioned cities are shown in figure 7 and figure 8. Figure 7 contains the values predicted by model 1 (epcohs:10) and Figure 8 contains the values predicted by model 2 (epochs:50).

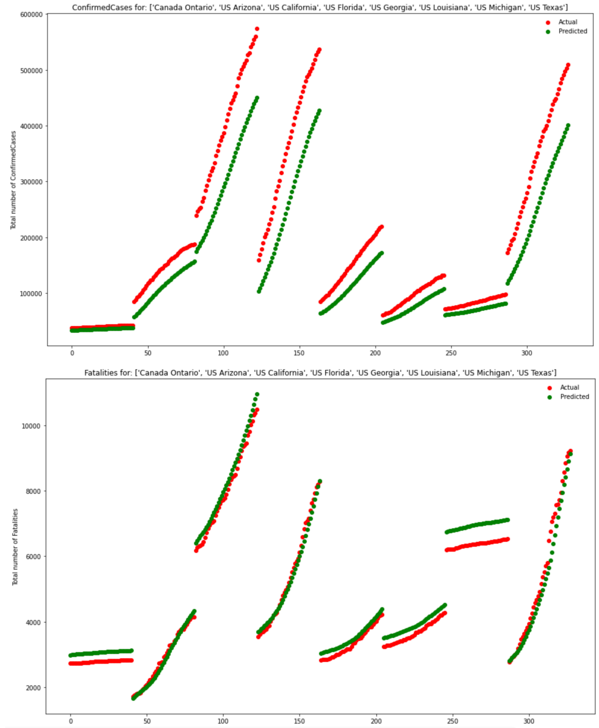

Fig. 7: (a) Actual & predicted confirmed cases for model 1, (b) Actual & predicted fatalities for model 1

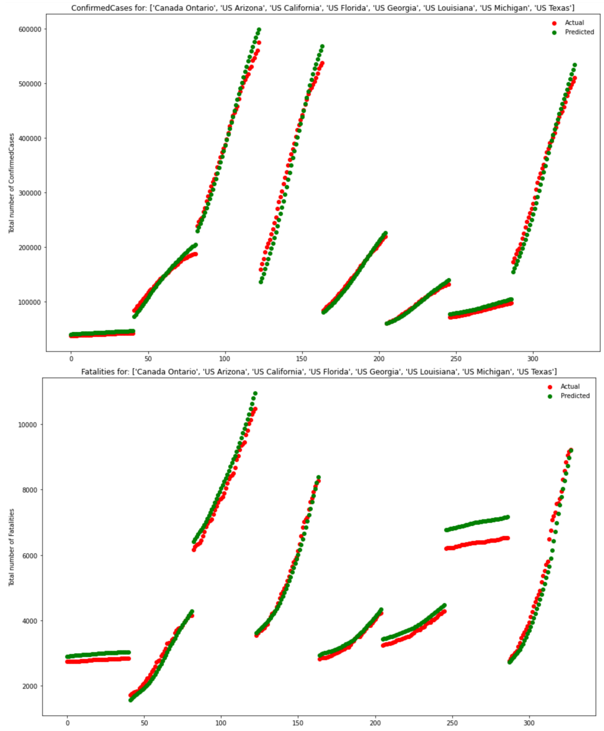

Fig. 8: (a) Actual & predicted confirmed cases for model 2, (b) Actual & predicted fatalities for model 2

We observe that for the aforementioned cities, model 2 (50 epochs) clearly outperforms model 1 for both confirmed cases and fatalities (10 epochs).

5. FUTURE SCOPE

Stacked LSTM model predicts the number of confirmed cases and fatalities quite well. However, the model does not perform well for countries like Brazil and India (as shown by the spikes in figure 9).

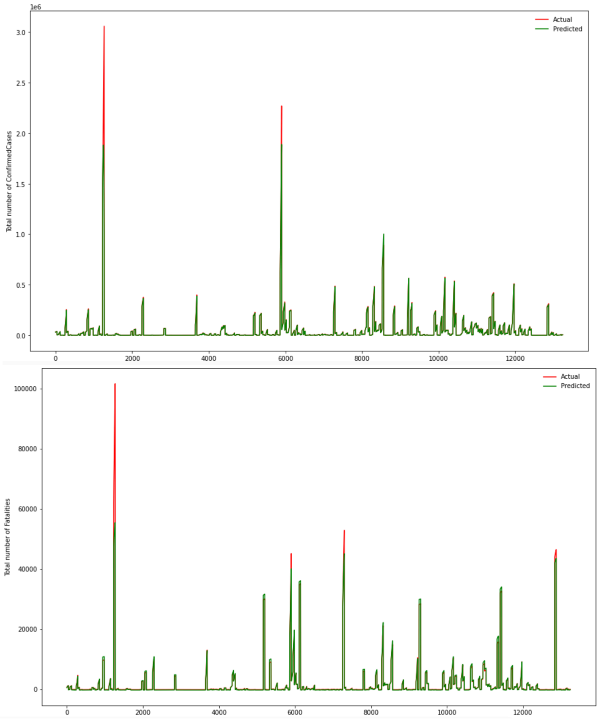

Fig. 9: Difference in actual and predicted values for (a) Confirmed Cases, (b) Fatalities

The model performs well when predicting regions with low COVID-19 cases, but there is a huge difference while predicting number of confirmed cases/fatalities in regions with high COVID-19 cases.

To overcome this issue, data of regions with high number of COVID-19 cases can be collected for each province/state. For example, US has the highest number of COVID-19 cases in the world, however, since the data for COVID-19 in US is collected for each state, we do not encounter such issue.

6. SUMMARY

In this paper, I built neural network (RNN) models to predict the number of confirmed cases and fatalities due to COVID-19. The 3-layer neural network comprised of Dense layer and stacked LSTM layers, and made prediction based on number of COVID-19 cases and fatalities in the last 10 days. I also trained the model on two different hyper-parameters and compared their performance using MSE and RMSE.

7. ACKNOWLEDGEMENT

I would like to extend my sincere gratitude to Professor Fathi Salem for giving me the opportunity and providing guidance throughout this project.

8. REFERENCE

[1] C. Olah, "Understanding LSTM Networks", Colah.github.io, 2015. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[2] "CSSEGISandData/COVID-19", GitHub, 2020. [Online]. Available: https://github.com/CSSEGISandData/COVID-19/

[3] I. Goodfellow, Y. Bengio and A. Courville, "Deep Learning", Deeplearningbook.org, 2016. [Online]. Available: https://www.deeplearningbook.org.

[4] N. Arbel, "How LSTM networks solve the problem of vanishing gradients", Medium, 2018. [Online]. Available: https://medium.com/datadriveninvestor/how-do-lstm-networks-solve-the-problem-of-vanishing-gradients-a6784971a577.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel and B. Thirion, "Scikit-learn: Machine Learning in {P}ython", Journal of Machine Learning Research, vol. 12, pp. 2825--2830, 2011.