Epidemic Forecasting for COVID 19 using Stacked CNN - Bidirectional LSTM Model

Author
Apoorv Jain
Btech CSE
PDPM IIITDM Jabalpur

Abstract

Coronavirus disease 2019 (COVID-19) is an infectious disease that emerged in China in December 2019 and has affected the whole world. On 30 January 2020, WHO declared it to be a Public International Health Emergency.The Coronavirus cases are still increasing in India also.We need to prepare forecasting models to assess the situation in the future,which can help in making the right decisions, concrete plans of action and restraining similar epidemics in the future.
In this study, the outbreak of this disease has been analysed and trained for Indian region till 10th May, 2020, and testing has been done for the number of cases for the next three weeks. It proposes a stacked LSTM model using CNN and Bidirectional LSTM layers which gives an RMSLE of 0.00925 on the test data of next three weeks.We hope that the present comparative analysis will provide an accurate picture of pandemic spread to the government officials so that they can take appropriate mitigation measures.

Keywords: COVID 19, Stacked LSTM, Time Series Forecasting, CNN, Bidirectional LSTM

1. Introduction

Access to accurate outbreak prediction models is essential to obtain insights into the likely spread and consequences of infectious diseases. Governments and other legislative bodies rely on insights from prediction models to suggest new policies and to assess the effectiveness of the enforced policies.The novel coronavirus disease (COVID-19) has been reported to have infected more than 138 million people, with more than 2.97 million confirmed deaths worldwide. The recent global COVID-19 pandemic has exhibited a nonlinear and complex nature.India's situation is not any better, with 14.1 million cases and 173 thousand deaths.
The role of data scientists and data mining researchers is to integrate the related data and technology to better understand the virus and its characteristics, which can help in making the right decisions and making concrete plans of action. It will lead to a bigger picture in taking aggressive measures in developing infrastructure, facilities, vaccines, and restraining similar epidemics in the future.

Machine learning and AI is a hope of light to assist in this situation.Formerly many epidemic forecasting models have been modelled like SIR,Polynomial Regression, Multilayer Perceptron, Grey wolf optimization, etc.

Journal	Outbreak Infection	Model
Transboundary and Emerging Diseases	Swine fever	Random Forest
Geospatial Health	Dengue fever	Dengue fever
Dengue fever	Influenza	Random Forest
Journal of Public Health Medicine	Dengue/Aedes	Bayesian Network
Water Research	Oyster norovirus	Genetic programming
Infectious Disease Modelling	Dengue	Classification and regression tree (CART)
Digital Government: Research and Practice	COVID 19	SEIR Compartment model

Fig 1 Outbreak and models proposed

2. Materials and Methods

2.1 Polynomial Regression model

Autoregression is a time series model that uses observations from previous time steps as input to a regression equation to predict the value at the next time step. It is a very simple idea that can result in accurate forecasts on a range of time series problems.AR basically means that it is a regression of itself. It can be performed with various degrees to find the best fit for the given data.

Autoregression of order p,

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \cdots + \beta_p Y_{t-p} + \epsilon_t$$

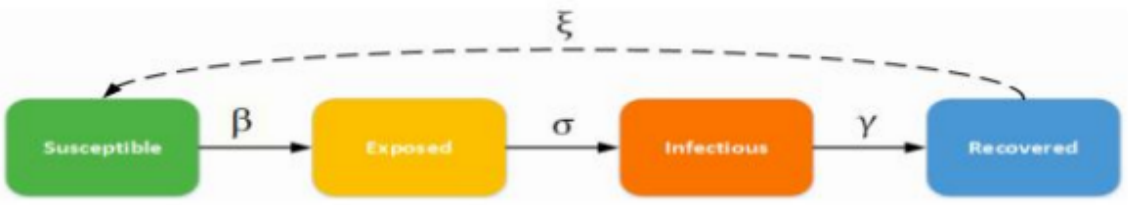
Order ‘p’ means, up to p-lags of Y is used.
α is the intercept
β1, β2 till βp are the coefficients of the lags of Y till the order p.

ε_{t} is the error, which is considered white noise.

2.2 SEIR model

SEIR(Suspected-Exposed-Infected-Recovered) is a compartment model based on statistical parameters. In this category of models, individuals experience a long incubation duration (the “exposed” category), such that the individual is infected but not yet infectious. For example, chicken-pox, and even vector-borne diseases such as dengue hemorrhagic fever have a long incubation duration where the individual cannot yet transmit the pathogen to others.

The SEIR model has mainly four components, viz., Susceptible (S), Exposed (E), Infected (I), and Recovered (R) .
S is the fraction of susceptible individuals (those able to contract the disease)
E is the fraction of exposed individuals (those who have been infected but are not yet infectious)
I is the fraction of infective individuals (those capable of transmitting the disease),
R is the fraction of recovered individuals (those who have become immune).



γ : the proportion of infected recovering per day.
β : expected amount of people coming in contact with an infected person per day.
σ : the proportion of exposed people getting infected per day.
ξ : the rate with which recovered individuals return to the susceptible state due to loss of immunity.

The model tries to predict the total number of people infected, and the duration of an epidemic, and to estimate various epidemiological parameters such as the reproductive number.

2.3 Neural Network methods

2.3.1 LSTM

A commonly used prediction strategy for time-series data (e.g. the epidemic data of daily cases that we consider here) is recurrent neural networks (RNNs). Specific types of RNNs that could provide robust prediction are LSTMs (Long Short-Term Memory units). LSTMs are able to recognize temporal patterns in time-series data that

are then used in the prediction.
Typical RNN uses information from the previous step to predict the output. But if only the previous step is not enough, that is long-term dependency. If we use RNN using all previous steps, the explosion/vanishing gradient problem is encountered.
LSTM can solve this problem because it uses gates to control the memorizing process. It has three gates:

- Input
- Forget, and
- Output

These gates are deciding the information to keep and discard according to its importance by using sigmoid and tanh activation.

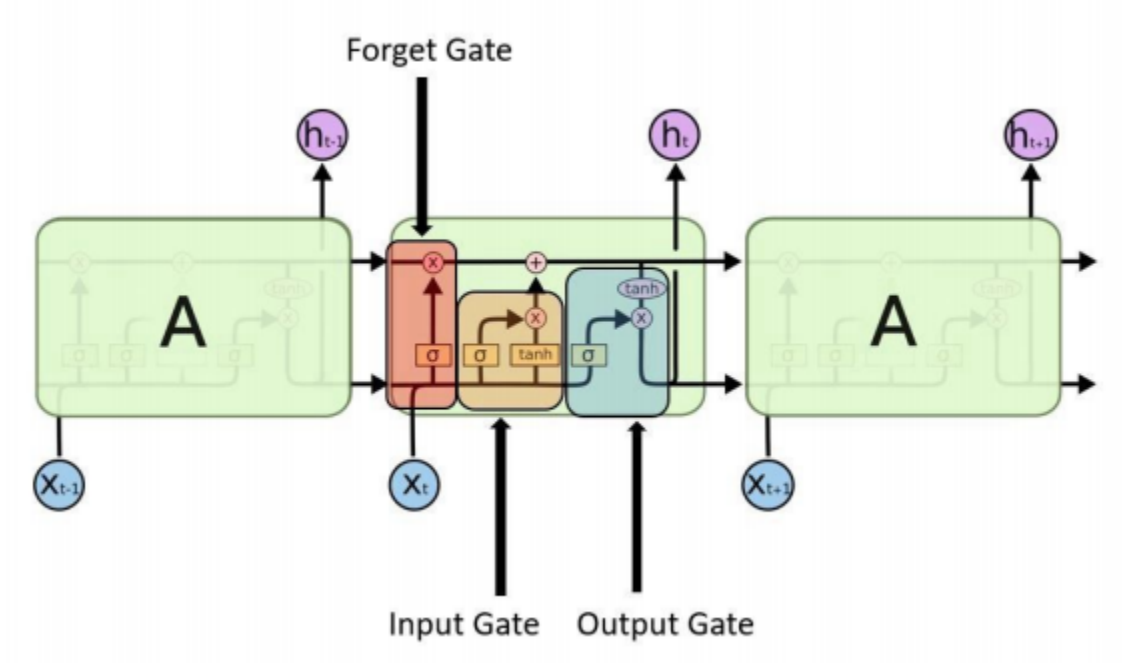


Fig 2 Working of LSTM

2.3.1 Standard Vanilla LSTM model

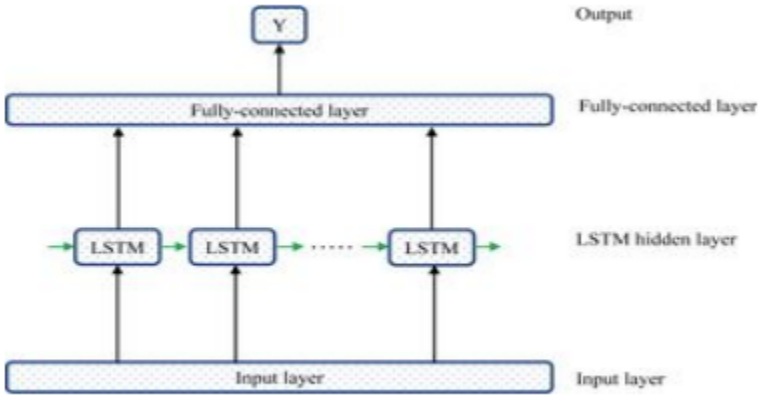


Fig 3. Vanilla LSTM Architecture

This LSTM architecture was designed into 4 layers: an input layer, an LSTM layer (hidden layer), a fully-connected layer, and an output. Each LSTM Layer have n neurons, and the activation function was ReLU. The loss function was MSE, andthe optimizer was “Adam”.

2.3.2 CNN LSTM

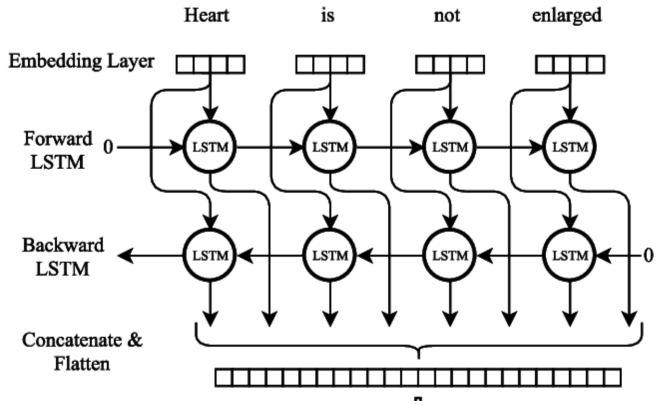
Convolutional Neural Network(CNN) are particular Deep Neural Network(DNN) based on the concept of weight sharing so that weights does not have to be as large as that for a fully connected structure. CNN contain generally four levels in structure: an input layer, convolutional layers, pooling layers, and fully connected layer (output). The con- volutional layer is the most important part of a CNN, in which the input is convoluted with several and each filter represents a smaller matrix and corresponding feature maps can be obtained after convolution operation. The pooling layer gives a summary statistic of the nearby outputs such as max-pooling and average-pooling, the most popular pooling layers, which outputs are respectively the maximum of a rectangular neighborhood and the aver- age of the rectangular neighborhood. The convolutional and pooling layers are generally used to extract features, and then one or more fully connected layers are usually adopted after one or more groups of convolutional and pooling layers. The fully connected layer can put the information from feature maps together, and then output them to latter layers.

To conclude, DNN are appropriate for mapping features to a more separable space, LSTM are good at temporal and times series modeling and CNN good at reducing frequency variations, so they are complementary in their modeling capabilities. This is used as an idea for CNN LSTM.

2.3.3 Bidirectional LSTM

A Bidirectional LSTM, or biLSTM, is a sequence processing model that consists of two LSTMs: one taking the input in a forward direction, and the other in a backwards direction. BiLSTMs

effectively increase the amount of information available to the network, improving the context available to the algorithm (e.g. knowing what words immediately follow and precede a word in a sentence).



Flg 4. Bidirectional LSTM working

2.3.4 Stacked CNN - Bidirectional LSTM

A Stacked LSTM architecture can be defined as an LSTM model comprises of multiple LSTM layers. An LSTM layer above provides a sequence output rather than a single value output to the LSTM layer below. Specifically, one output per input time step, rather than one output time step for all input time steps.

The main reason for stacking LSTM is to allow for greater model complexity. In case of a simple feedforward net we stack layers to create a hierarchical feature representation of the input data to then use for some machine learning task. The same applies for stacked LSTM,at every time step an LSTM, besides the recurrent input. If the input is already the result from an LSTM layer (or a feedforward layer) then the current LSTM can create a more complex feature representation of the current input.

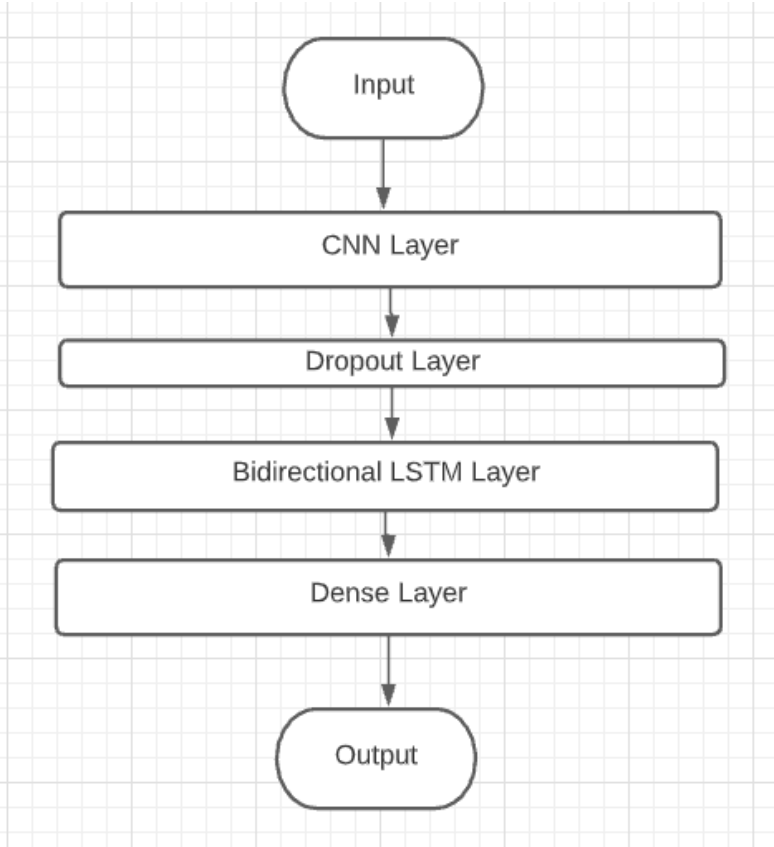


Fig 5. Proposed CNN Bidirectional Stacked model

This is a stacked LSTM model which consists of CNN, Dropout,Bidirectional and Dense Layer.The model takes advantage of feature detection ability of CNN and contextual boost of bidirectional LSTM to discover complex pattern easily. Dropout Layer is used for avoiding overfitting of the model.Dense Layer finally summarizes all the information learned by the model.

2.2 Dataset

The dataset is extracted from the Github Repository of John Hopkins University USA .
It is shrinked to data for indian regions.The time period of training data is from 30/01/2020 to 10/05/2020, and the test data from the time period 11/05/2020 to 31/05/2020.It consists of two columns: Date and cumulative cases on that day.

2.3 Training

The training was performed on Google Colab Platform in Python language.The Stacked CNN-Bidirectional LSTM model was trained for 200 epochs with 64 filters in CNN layer and 50 neurons in Bidirectional Layer.The dataset was transformed for 2 timestep lag,The model was

trained on 70 instances and validated on 30 instances of the data.This achieved a RMSLE of **0.00925**.

2.4 Evaluation criteria

The Root Mean Squared Log Error (RMSLE) can be defined using a slight modification on sklearn's mean_squared_log_error function, which itself a modification on the familiar Mean Squared Error (MSE) metric.

The formula for RMSLE is represented as follows:

$$\text{RMSLE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i+1)-\log(a_i+1))^2}$$

Where:

n is the total number of observations in the (public/private) data set,

p_i is your prediction of target, and

a_i is the actual target for i.

$log(x)$ is the natural logarithm of x ($log_e(x)$.

3. Results

The stacked LSTM performed quite better than SEIR,Polynomial Regression,Vanilla LSTM,CNN LSTM and Bidirectional LSTM. Vanilla LSTM model performance can be improved on adding dropout layers but still the stacked model outperform it.

RMSLE of Polynomial Regression 1.75
RMSLE of SEIR: 1.52
RMSLE of Vanilla LSTM: 0.542
RMSLE of Bidirectional LSTM :0.2
RMSLE of CNN LSTM :0.12
RMSLE of Stacked CNN Bidirectional LSTM :0.00925

	Date	Standard Vanilla LSTM	Bidirectional	CNN LSTM	Stacked CNN and Bidirecctional	Actual
0	5/13/20	108997.038194	72991.831628	72991.831628	78118.767305	78055.0
1	5/14/20	116581.909357	77219.498597	77219.498597	82014.549576	81997.0
2	5/15/20	124805.138570	81715.017510	81715.017510	86092.110253	85784.0
3	5/16/20	133202.508374	86098.018752	86098.018752	90005.826179	90648.0
4	5/17/20	143145.719191	91819.679256	91819.679256	95027.289549	95698.0
5	5/18/20	154640.215164	97869.501625	97869.501625	100234.262167	100328.0
6	5/19/20	166055.851093	103513.743182	103513.743182	105002.100426	106475.0
7	5/20/20	179632.332302	111150.878139	111150.878139	111322.745000	112028.0
8	5/21/20	194265.922400	118190.101844	118190.101844	117023.053246	118226.0
9	5/22/20	209730.705372	126202.946869	126202.946869	123374.281107	124794.0
10	5/23/20	226946.837831	134872.331787	134872.331787	130091.243440	131423.0
11	5/24/20	245182.002188	143806.446272	143806.446272	136855.938116	138536.0
12	5/25/20	264830.050322	153596.540504	153596.540504	144097.547918	144950.0
13	5/26/20	284347.326885	162603.334328	162603.334328	150611.819921	150793.0
14	5/27/20	302460.510365	170954.060617	170954.060617	156532.696110	158086.0
15	5/28/20	322913.619476	181569.583336	181569.583336	163904.324909	165386.0
16	5/29/20	345638.858601	192406.250435	192406.250435	171261.558748	173491.0
17	5/30/20	370400.720421	204681.020514	204681.020514	179404.078508	181827.0
18	5/31/20	397181.175209	217567.376257	217567.376257	187749.024397	190609.0

Fig 6. Prediction of different models for next three weeks

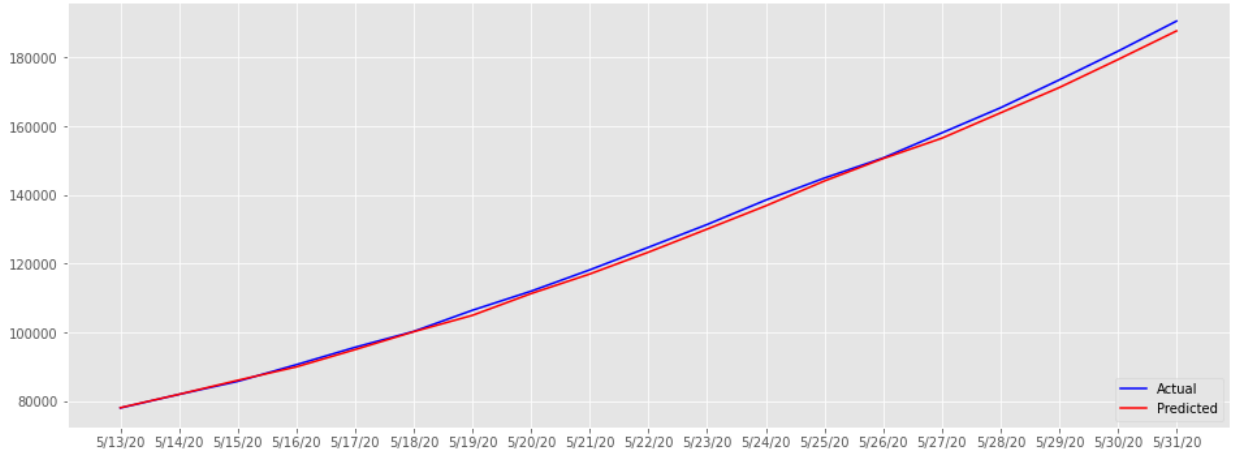


Fig 7. Plot of actual and predicted for next three weeks

4. Conclusion

Forecasting epidemic is very crucial to assess the future situation and make decisions accordingly.AI can play a major role in forecasting COVID 19 in the present times.Our model predicted very well on the dataset using stacked LSTM approach.So the deep learning frameworks like LSTM can perform very well on the when optimized hyperparameters are used. The various features like lockdown,recovery rate and vaccination rate can further boost the accuracy of the model.

5. Acknowledgment

This work was done under the guidance of Dr. Kusum Kumari Bharti, Assistant Professor in Computer Science and Engineering Department at Indian Institute of Information Technology, Design, and Manufacturing, Jabalpur. I also thank Mr. Yash Shah and Mr. Shubham Jhanwar for contributing to research and development work.

6. References

[1] RAJAN GUPTA,GAURAV PANDEY and POONAM CHAUDHARY,SAIBAL K. PAL - Machine Learning Models for Government to Predict COVID-19 Outbreak(paper)
[2] Fenglin Liu ,Jie Wang,Jiawen Liu,Yue Li,Dagong Liu- Predicting and analyzing the COVID-19 epidemic in China: Based on SEIRD, LSTM and GWR models(paper)
[3] Hafiz Tayyab Rauf, M. Ikram Ullah Lali, Muhammad Attique Khan, Seifedine Kadry, Hanan Alolaiyan, Abdul Razaq & Rizwana Irfan(paper)
[4]Sourabh Shastri, Kuljeet Singh, Sachin Kumar, Paramjit Kour & Vibhakar Mansotra Deep-LSTM ensemble framework to forecast Covid-19: an insight to the global pandemic(paper)
[5] Liang, R.; Lu, Y.; Qu, X.; Su, Q.; Li, C.; Xia, S.; Liu, Y.; Zhang, Q.; Cao, X.; Chen, Q.; et al. Prediction for global African swine fever outbreaks based on a combination of random forest algorithms and meteorological data. Transbound. Emerg. Dis. 2020, 67, 935–946.(paper)
[6] Anno, S.; Hara, T.; Kai, H.; Lee, M.-A.; Chang, Y.; Oyoshi, K.; Mizukami, Y.; Tadono, T. Spatiotemporal dengue fever hotspots associated with climatic factors in Taiwan including outbreak predictions based on machine-learning. Geospat. Health 2019, 14, 183–194.
[7]Tapak, L.; Hamidi, O.; Fathian, M.; Karami, M. Comparative evaluation of time series models for predicting influenza outbreaks: Application of influenza-like illness data from sentinel sites of healthcare centers in Iran. BMC Res. Notes 2019, 12, 1–6
[8] Wenjie Lu,[1,2] Jiazheng Li,[3] Yifan Li,[3] Aijun Sun,[1] and Jingyang Wang A CNN-LSTM-Based Model to Forecast Stock Prices(paper)
[9] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," in *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673-2681, Nov. 1997, doi: 10.1109/78.650093.(paper)
[10] Rohitash Chandraa , Ayush Jainb , Divyanshu Singh Chauhanc Time series forecasting of COVID-19 transmission in Asia Pacific countries using deep neural networks(paper)