COVID-19 forecasting in India: A comparison between different Artificial Neural Network approaches

Data Science in the Life Sciences - Summer 2021

Outline

1. The Paper
2. Experimental Data
3. Data Preprocessing
4. Methods (MLP, CNN, LSTM, GRU)
5. Experimental Setup
6. Results
7. Conclusion
8. Discussion I
9. Data Exploration
10. Discussion II

The Paper

- time series forecasting of COVID-19 cases
- comparison of different artificial neural network approaches
- proposal of a Convolutional Neural Network (CNN) for the prediction
- evaluation of the efficacies of the different algorithms

Multiple-Input Deep Convolutional Neural Network Model for COVID-19 Forecasting in China

Chiou-Jye Huang [1], Yung-Hsiang Chen [2], Yuxuan Ma [1] and Ping-Huan Kuo [3,*]

[1] School of Electrical Engineering and Automation, Jiangxi University of Science and Technology, Ganzhou, Jiangxi, 341000, China; cjhuang@jxust.edu.cn; 1103552453@qq.com
[2] Department of Mechanical Engineering, National Pingtung University of Science and Technology, Pingtung 91201, Taiwan; yhchen@mail.npust.edu.tw
[3] Computer and Intelligent Robot Program for Bachelor Degree, National Pingtung University, Pingtung 90004, Taiwan; phkuo@mail.nptu.edu.tw
* Correspondence: phkuo@mail.nptu.edu.tw; Tel.: +886-87663800 ext. 32620

Abstract: COVID-19 is spreading all across the globe. Up until March 23, 2020, the confirmed cases in 173 countries and regions of the globe had surpassed 346,000, and more than 14,700 deaths had resulted. The confirmed cases outside of China had also reached over 81,000, with over 3,200 deaths. In this study, a Convolutional Neural Network (CNN) was proposed to analyze and predict the number of confirmed cases. Several cities with the most confirmed cases in China were the focus of this study, and a COVID-19 forecasting model, based on the CNN deep neural network method, was proposed. To compare the overall efficacies of different algorithms, the indicators of mean absolute error and root mean square error were applied in the experiment of this study. The experiment results indicated that compared with other deep learning methods, the CNN model proposed in this study has the greatest prediction efficacy. The feasibility and practicality of the model in predicting the cumulative number of COVID-19 confirmed cases were also verified in this study.

Keywords: Total confirmed forecasting; convolutional neural network; COVID-19

The Paper - Data

- Country: China
  - seven Chinese cities
- Time period: really short!
  - ~ 1 month and a few days

- Features:
  - Cumulative confirmed cases
  - New confirmed cases
  - Cumulative cured cases
  - New cured cases
  - Cumulative deaths
  - New deaths

The Paper - Results

Evaluation Metric	Method	Avg. Low Error	Avg. High Error
Mean Absolute Error	**CNN**	**102.943**	**284.573**
$\text{MAE} = \frac{1}{N}\sum_{n=1}^{N}\lvert y_n - \hat{y}_n\rvert$	MLP	5710.293	5811.429
	LSTM	2992.976	3324.591
	GRU	2599.150	1916.904
Root Mean Square Error	**CNN**	**109.439**	**325.857**
$\text{RMSE} = \sqrt{\frac{\sum_{n=1}^{N}(y_n - \widehat{y_n})^2}{N}}$	MLP	5715.125	5816.079
	LSTM	2994.851	3331.925
	GRU	2602.666	1924.361

The Project

Experimental Data

- Data taken from: https://github.com/covid19india/api
- Five most affected Indian states
- Time period: 10 March 2020 - 30 June 2021
- Training data (80%): 10 March 2020 - 28 March 2021
- Test data (20%): 29 March 2021 - 30 June 2021
- Contains information on cumulative features
- Short time period: 10 March 2020 - 18 June 2020

Data Preprocessing

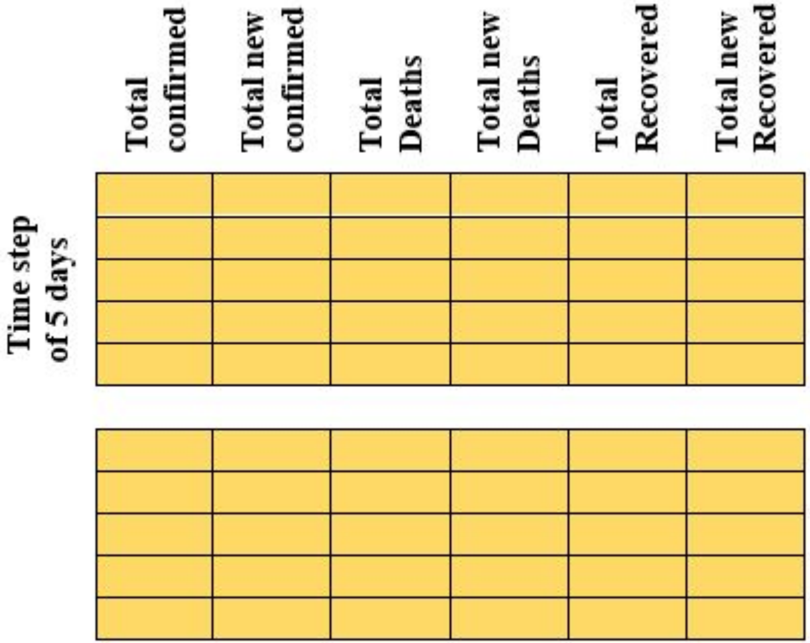
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import pyplot
import numpy as np
from sklearn.preprocessing import MinMaxScaler
```

- India dataset
- The most affected states
- Check the time period
- Missing values
- Compute the new cases (confirmed, deaths, and recovered)
- Data exploration
- Split the data train & test (80 % - 20 %)
- Data normalization (MinMaxScaler)

Data Preprocessing - Reshape

[Samples, ,]

Data Preprocessing - Reshape

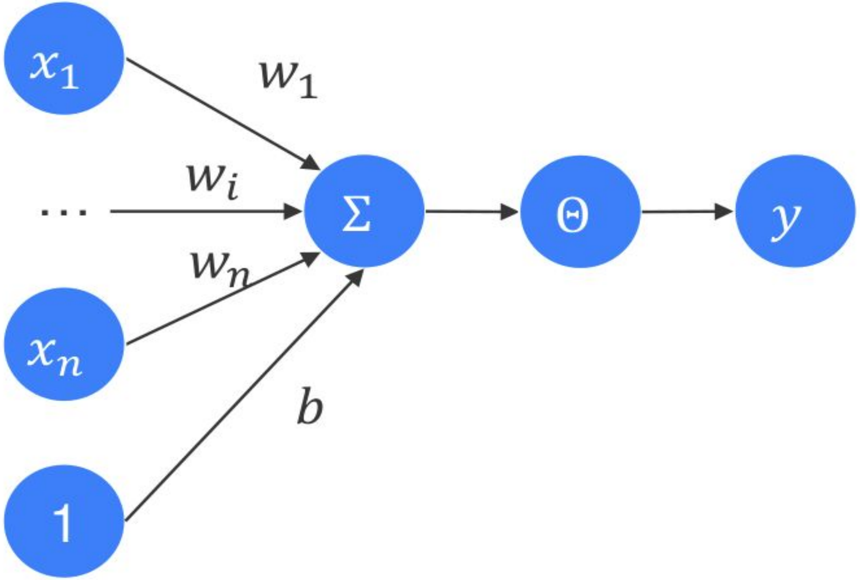


[Samples, Time step, Features]

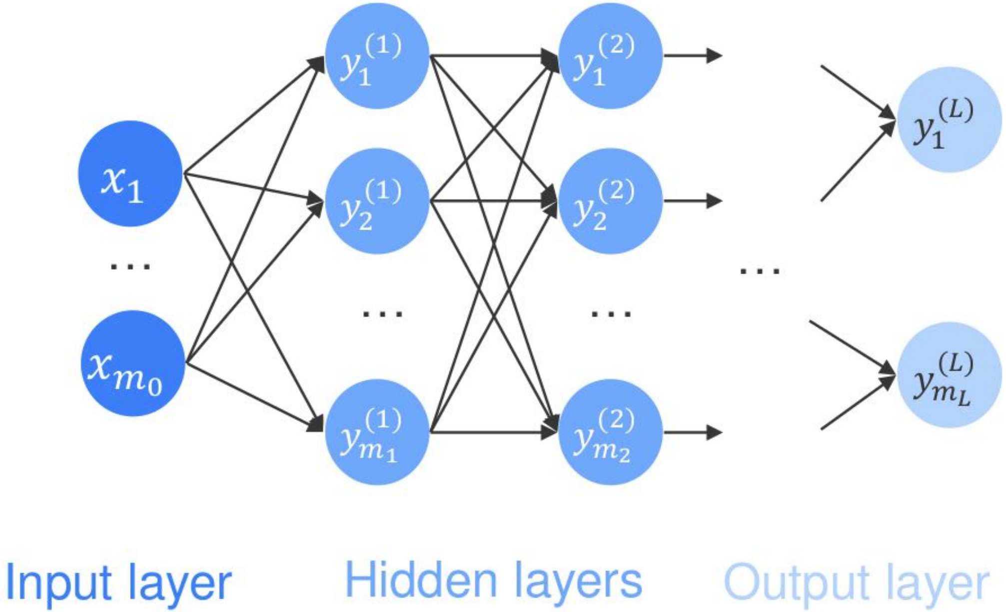
Methods

Multilayer Perceptron(MLP)

Normal Perceptron

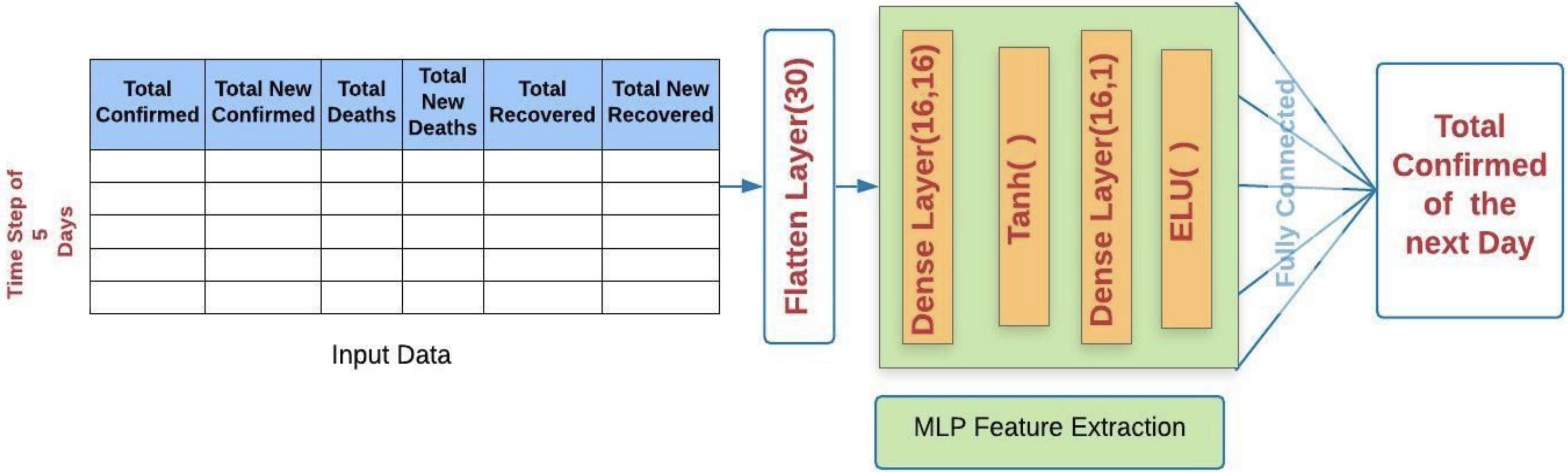


Multilayer Perceptron(MLP)

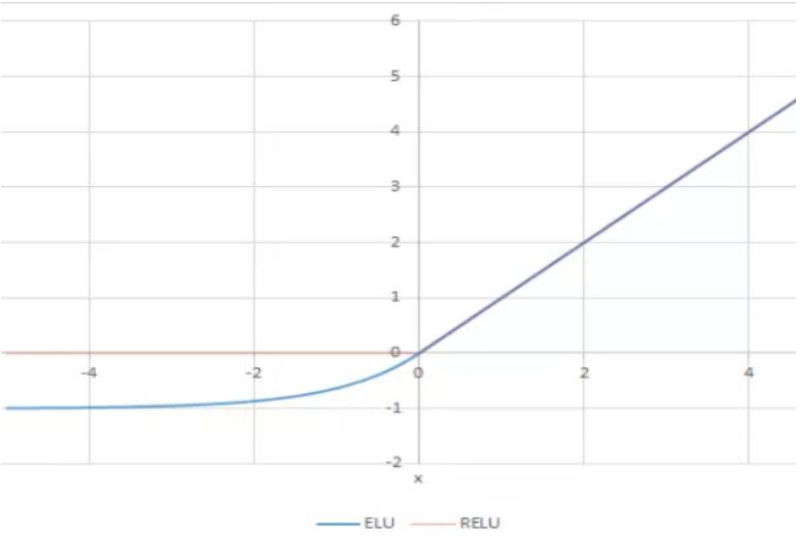


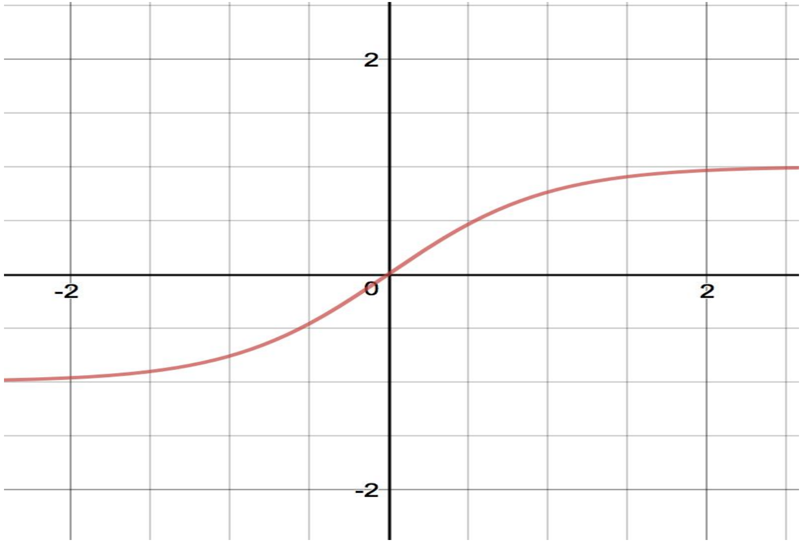
Implemented Architecture of MLP

- Parameters:
  - `Optimizer: Stochastic Gradient Descent (Learning rate= 0.001, momentum=0.9)`
  - Number of Epochs: 100, Batch Size: 16



Activation Functions (ELU and Tanh):



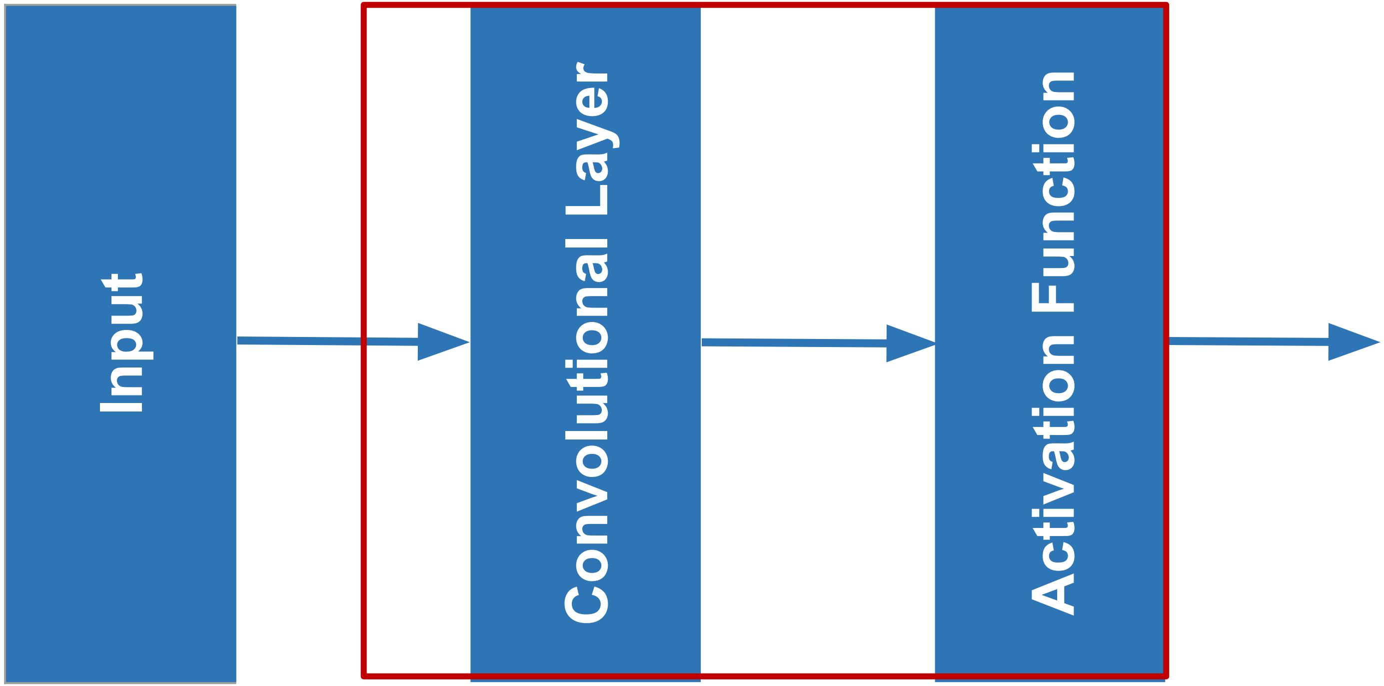


$$ELU(z) = \begin{cases} z, z > 0 \\ \alpha.(e^z - 1) z > 0, z <= 0 \end{cases}$$

$$Tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

pytorch [15]

Convolutional Neural Network (CNN) Block:

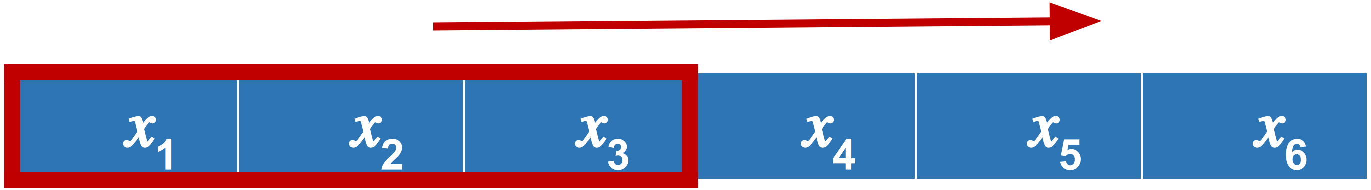


Convolutional Layer

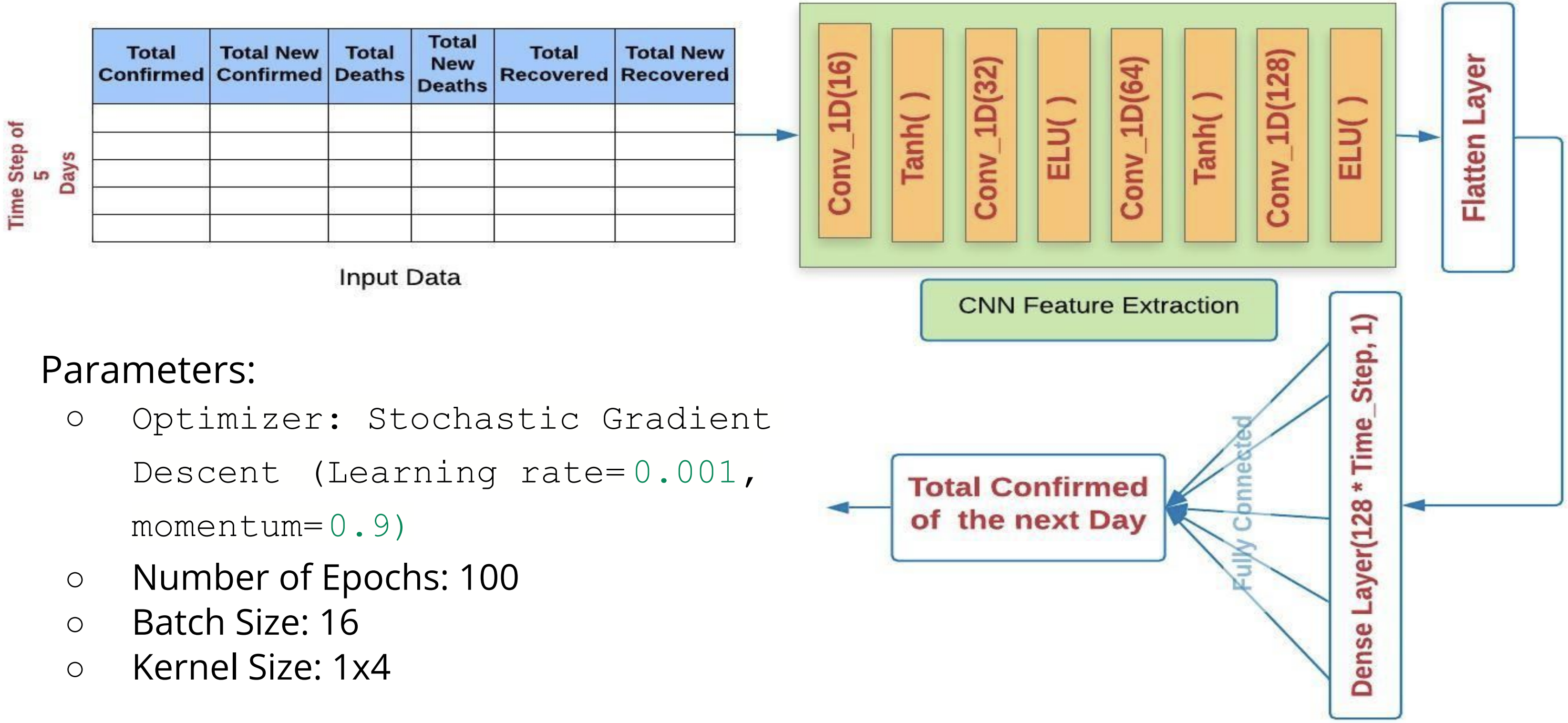
- 1D Convolution
- It is a linear operation
- Mathematical Definition:

For $x \in \mathbb{R}^n$, $K \in \mathbb{R}^m$ and $m < n$

$$(x * K)_i = \sum_{j=1}^{m} x_{i+j-1} K_j$$



Implemented CNN Architecture

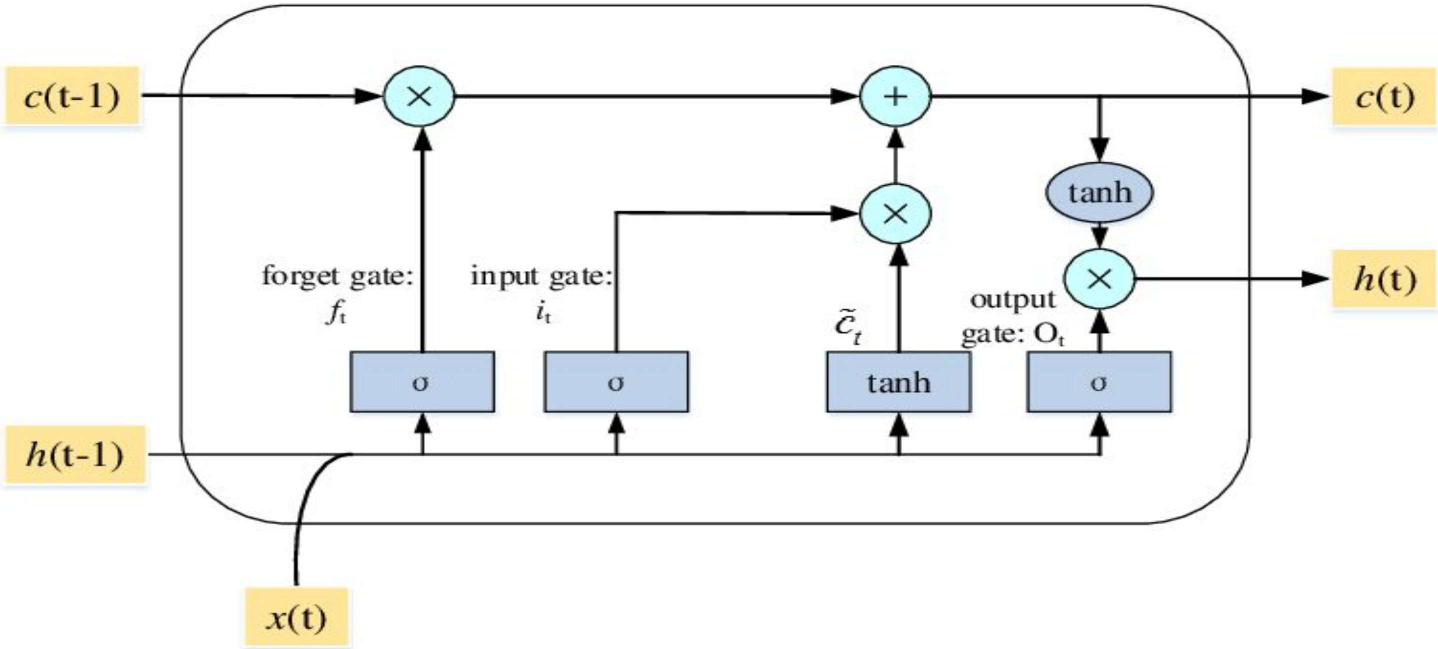


- Parameters:
  - `Optimizer: Stochastic Gradient Descent (Learning rate=0.001, momentum=0.9)`
  - Number of Epochs: 100
  - Batch Size: 16
  - Kernel Size: 1x4

Long Short Term Memory (LSTM)

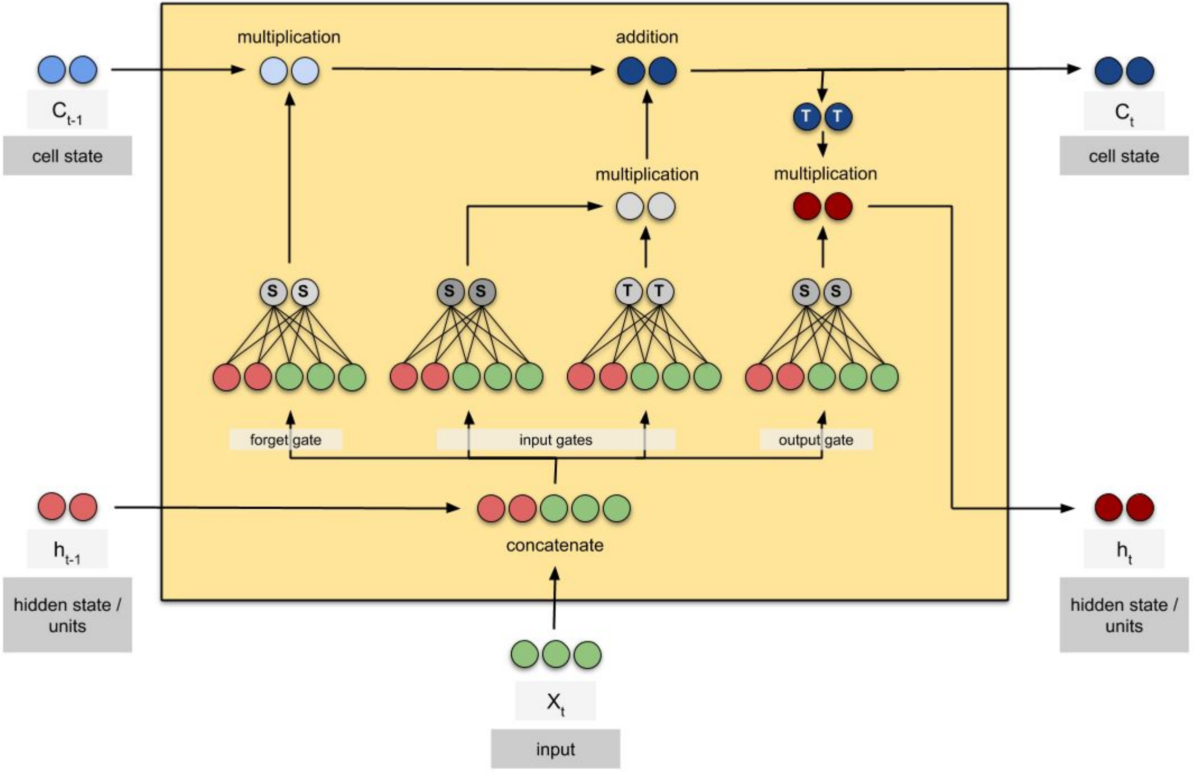
- Special type of Recurrent Neural Network (RNN)
- Motivation: It solves the Vanishing/Exploding problem of RNN.
- Capable to store long term dependencies
- Contain an extra cell comparing to RNN
- It contains:
    - Input gate layer
    - Forget gate layer
    - Output gate layer

LSTM Unit Architecture



Yuan et al. [13]

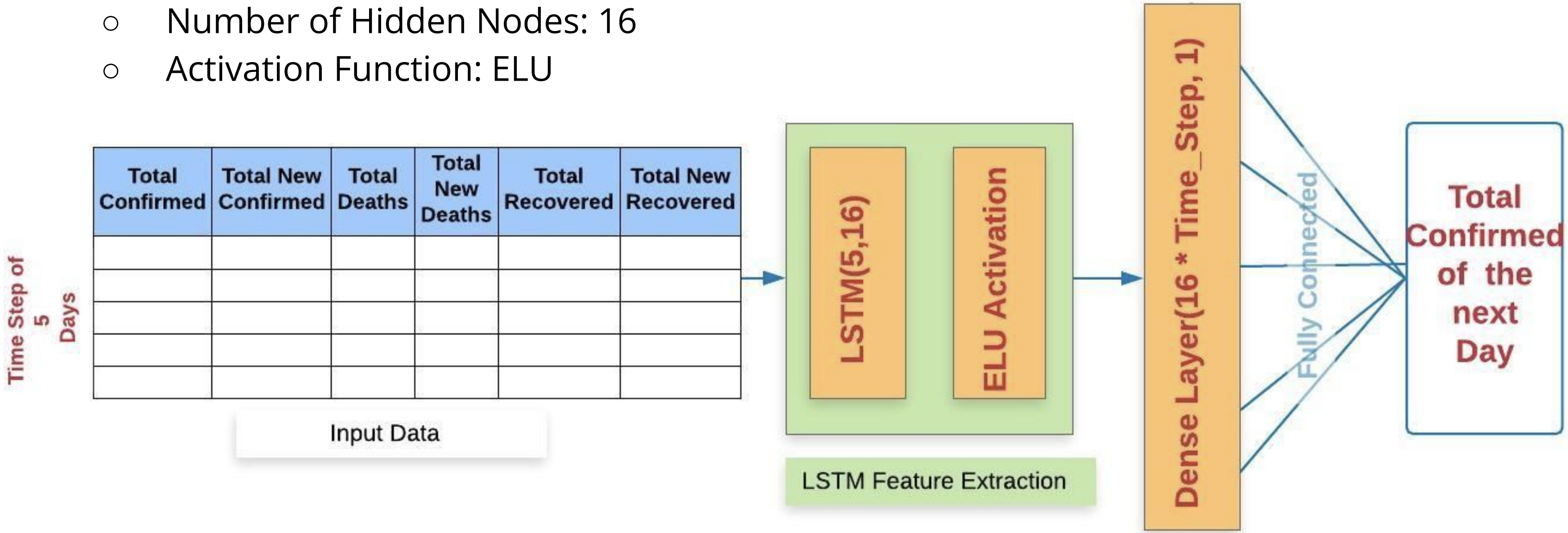
LSTM: Working Flow



Karim . [16]

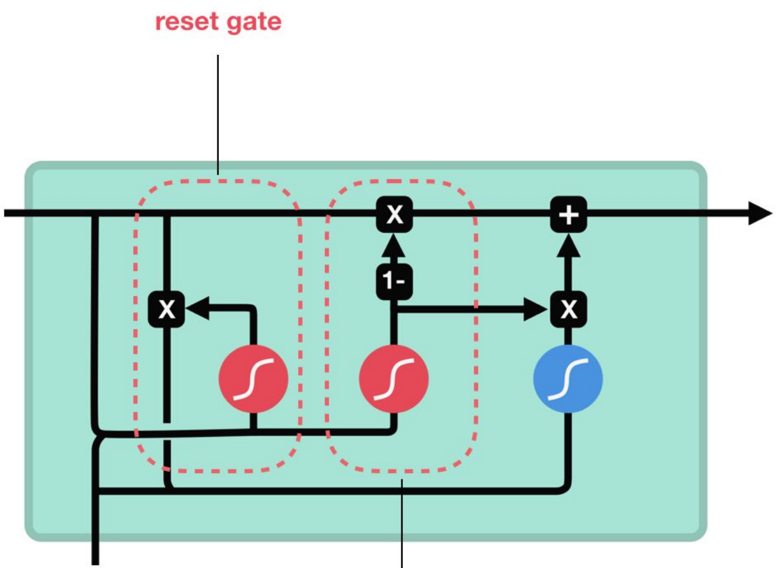
Implemented LSTM architecture

- Parameters:
  - Optimizer: Stochastic Gradient Descent (Learning rate= 0.001, momentum=0.9)
  - Number of Epochs: 100
  - Number of Hidden Nodes: 16
  - Activation Function: ELU



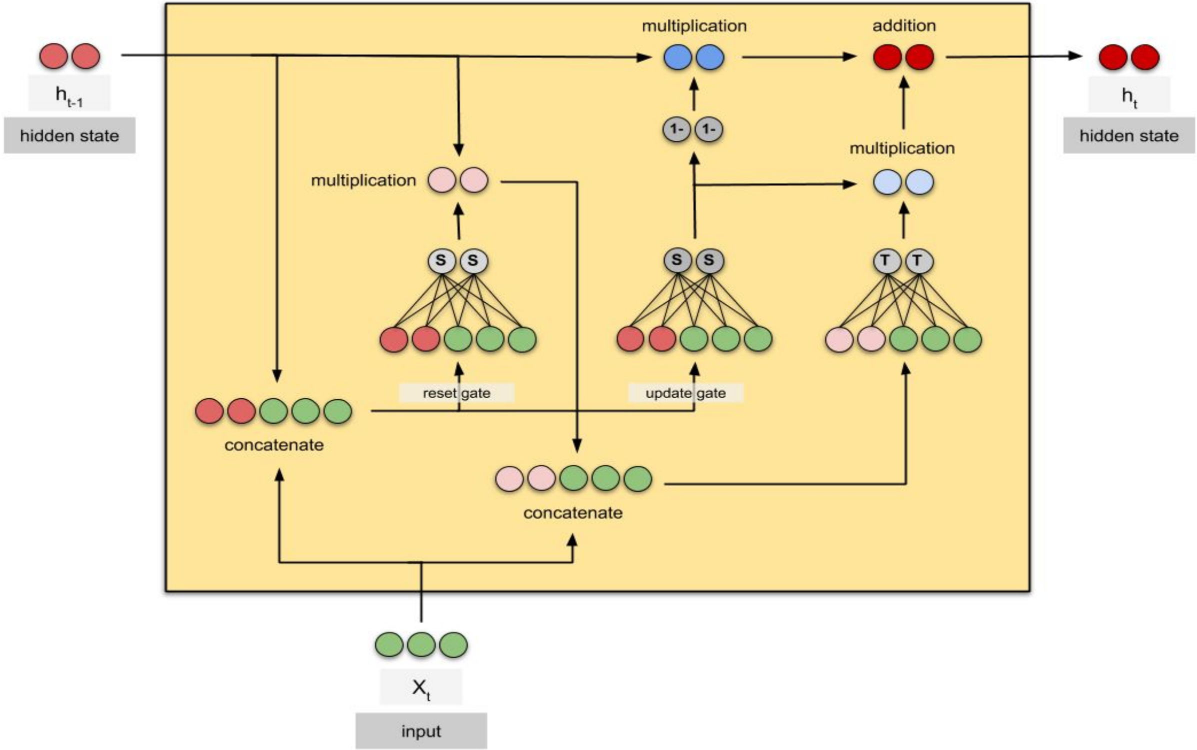
Gate Recurrent Unit - GRU

- Simplified Version of LSTM.
- Combines the input gate and the forget gate into the update gate.
- Uses less parameters comparing to LSTM



Phi et al. [14]

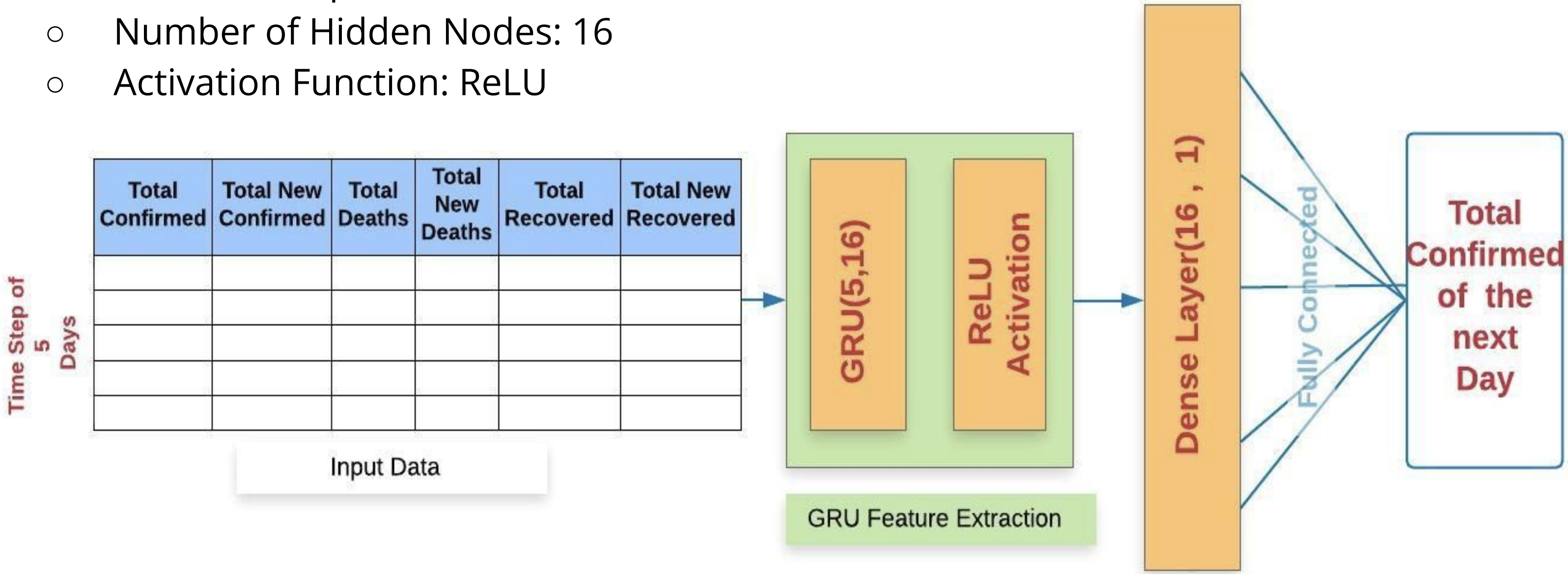
Gate Recurrent Unit - GRU



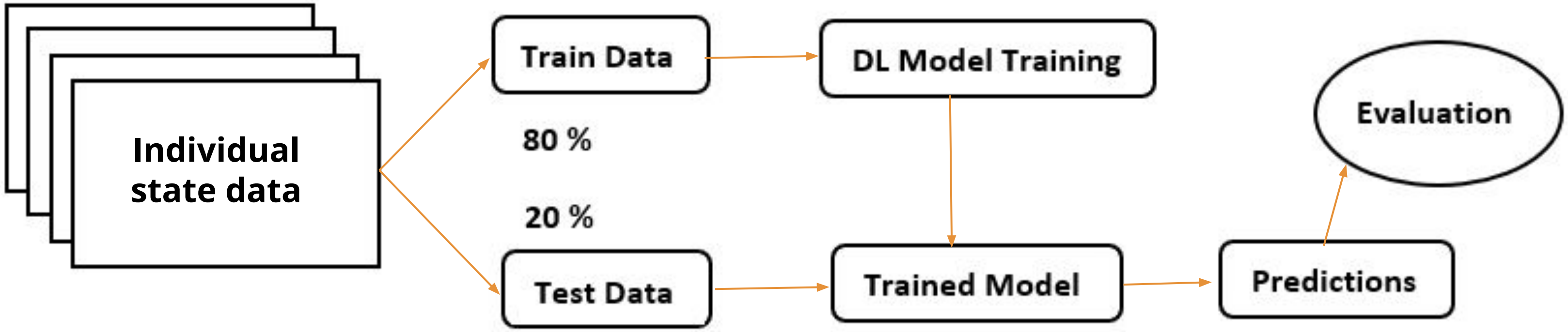
Karim . [16]

Implemented GRU architecture

- Parameters:
  - `Optimizer: Stochastic Gradient Descent (Learning rate= 0.001, momentum=0.9)`
  - Number of Epochs: 100
  - Number of Hidden Nodes: 16
  - Activation Function: ReLU



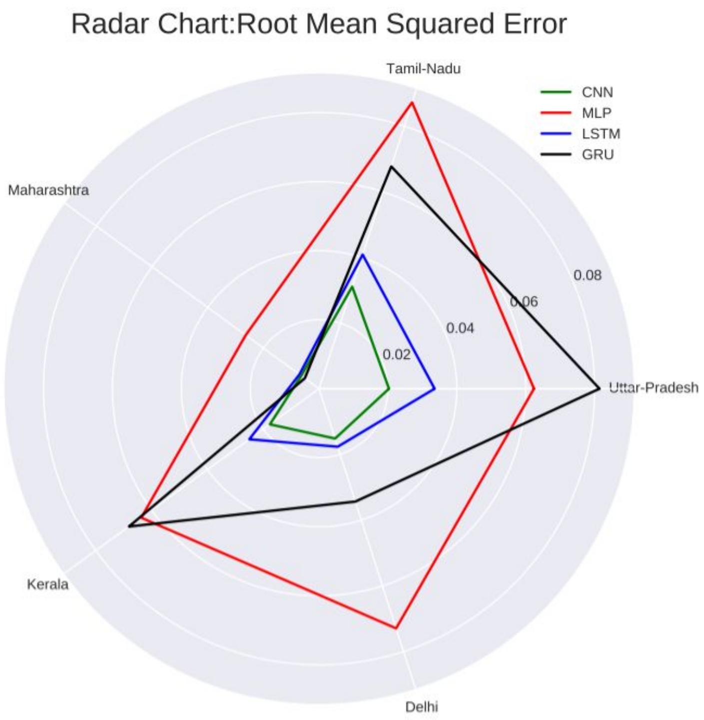
Experimental Setup



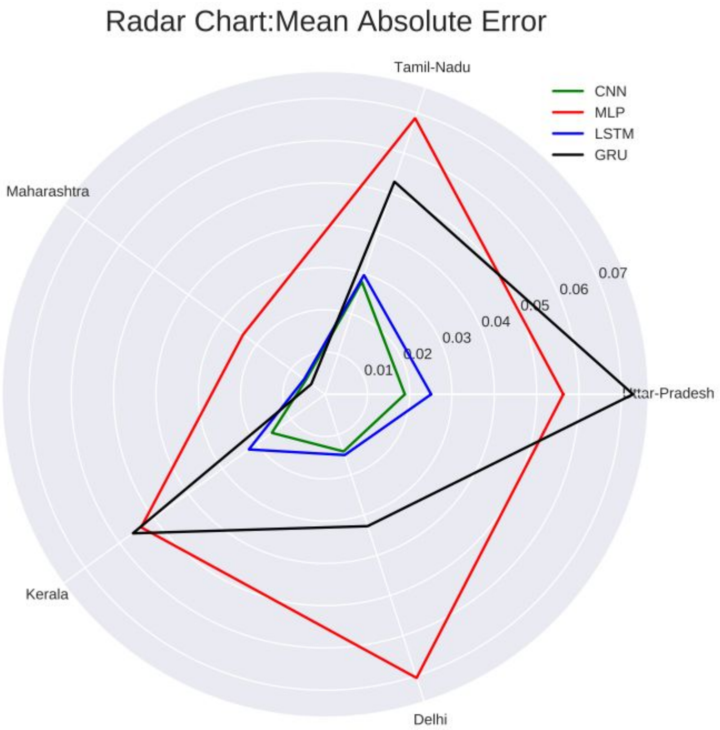
Technology: Python (pytorch, scikit-learn, matplotlib)

Results & Discussion

Results

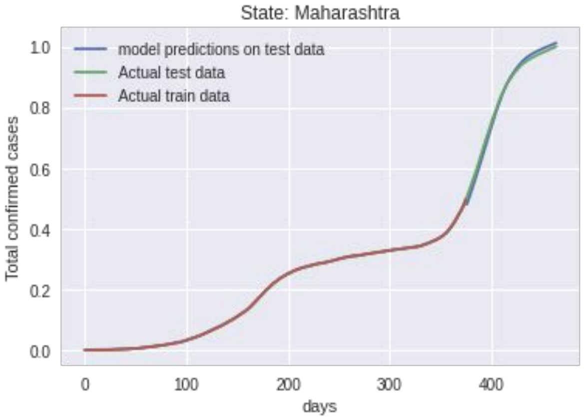


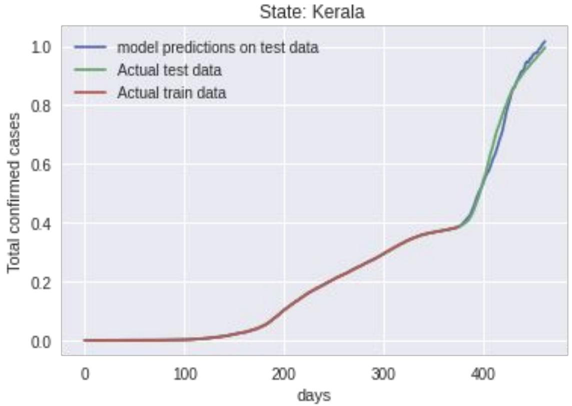
Radar Chart:Root Mean Squared Error
Tamil-Nadu
Maharashtra
Kerala
Delhi
Uttar-Pradesh
0.02
0.04
0.06
0.08
CNN
MLP
LSTM
GRU

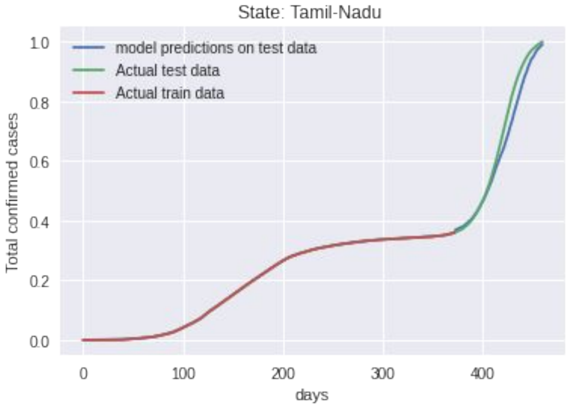


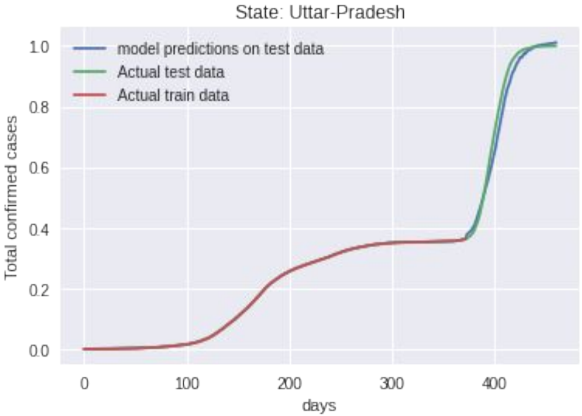
Radar Chart:Mean Absolute Error
Tamil-Nadu
Maharashtra
Kerala
Delhi
Uttar-Pradesh
0.01
0.02
0.03
0.04
0.05
0.06
0.07
CNN
MLP
LSTM
GRU

CNN Predictions









Conclusion

Different time periods

Different number of features

Different time steps

Conclusion

Different time periods

Different number of features

Different time steps

- The proposed CNN model is superior to those of its counterparts in terms of MAE and RMSE
- Increasing the time step:
    - improved the accuracy of CNN model
    - decreased the accuracy of the other models
- If we increase the number of input features the efficiency of the models decreases

Discussion I

Factors that influence the performance of DL methods:

- The data
  - Data amount
  - The relationship between the features
- The structure
  - Number of layers
  - Hyperparameters tuning
- Using hybrid models (Dutta et. al. [3])
- Methods comparison is recommended in practice

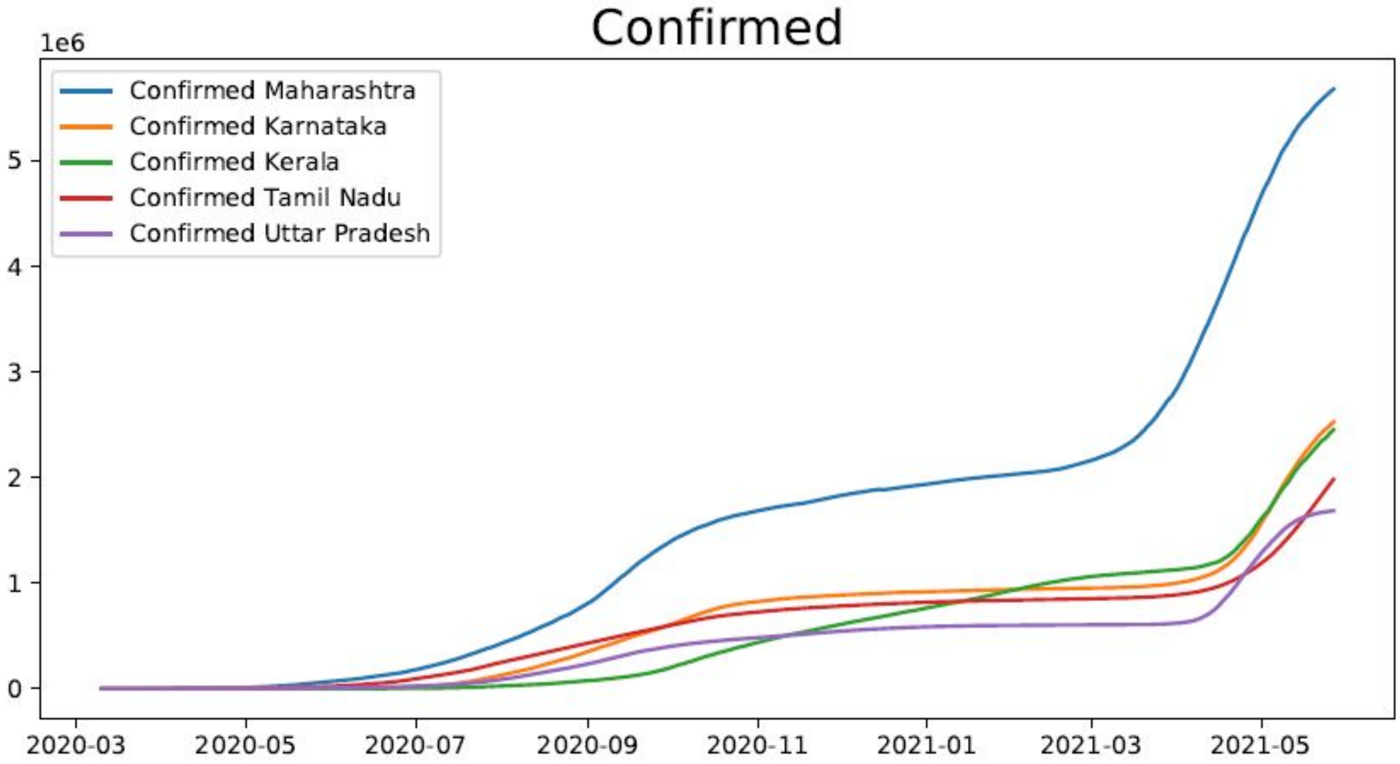
Our experimental results

The seed paper

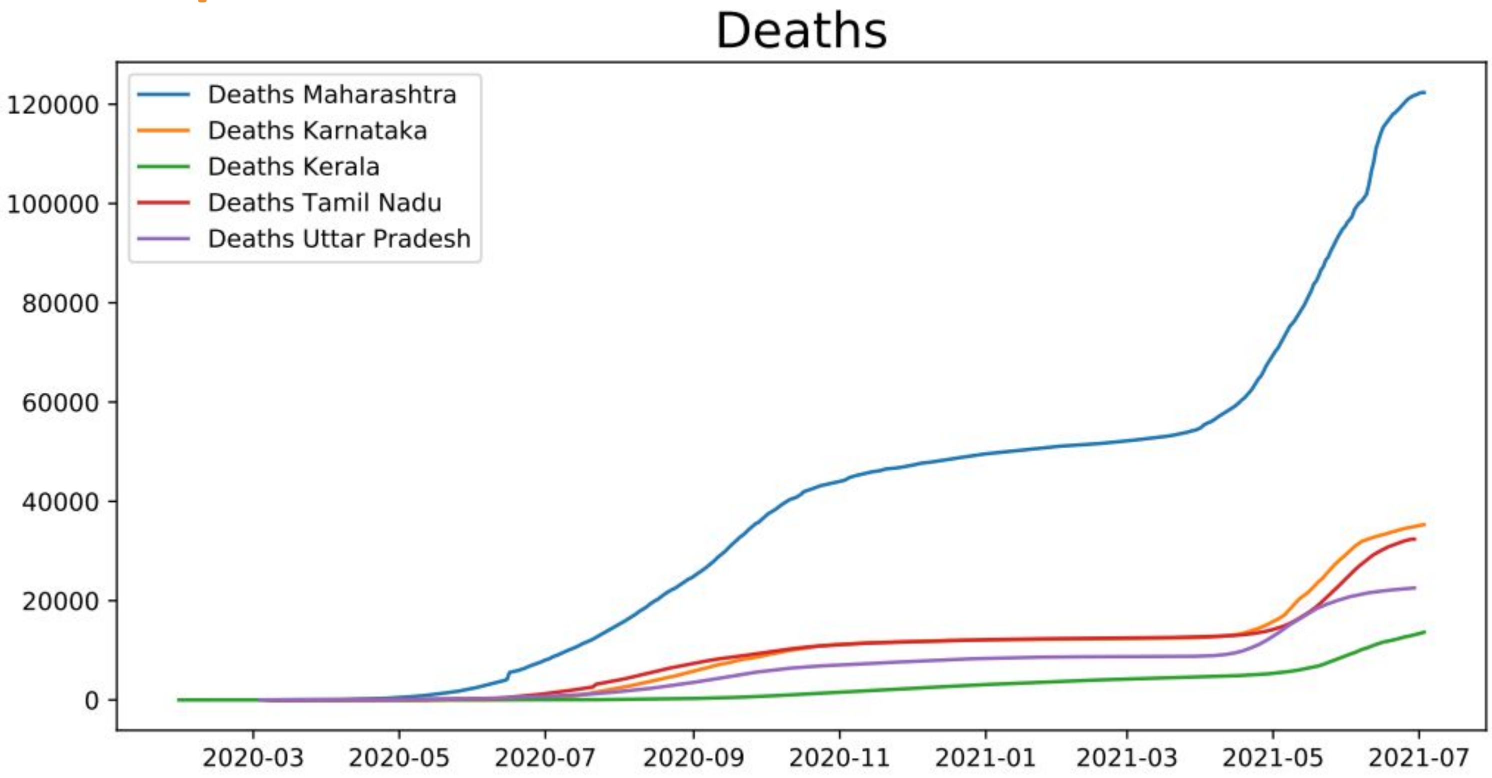
The related work

GitHub: https://github.com/sureshkuc/Data-Science-in-Life-Science-Project

Data Exploration



Data Exploration



Discussion II

State	Location	Population density	Health care	Social Events	Government interventions
Kerala	South, Malabar coast	**860 / km²**	robust during 1st wave, can't keep up since 2nd wave, best **testing ratio**	**Elections** in April 2021	**Contact tracing** and targeted **tests**
Uttar Pradesh	North, Taj Mahal	**828 / km²**	**Bad conditions** in hospitals, vaccine **shortages**	**Holi festival**, Migrant workers crisis	Contact tracing Closure of educational institutions, **Lockdown**
Tamil Nadu	South, Hindu temples	550 / km²	**85 labs** for tests, >75k beds	**Religious congregation event**	Contact tracing, testing, surveillance, **medical supplies**
Maharashtra	West/Central, Mumbai	370 / km²	**High testing rate**	**Mobility**, **Religious events**, **mass flouting of covid norms**	Contact tracing, targeted tests
Karnataka	South West, Arabian Sea	319 / km²	~10k tests/day, **74 labs**, >21k beds reserved	**Religious events**	Early containment efforts, **Closure of borders** with Kerala, Lockdown

Project History

- Working:
  - Github account with shared access
  - Code: Google Colaboratory (shared)
  - Presentation: Google slides (shared)
  - Report: Overleaf (shared)
- Failures and bad experiences:
  - Misunderstandings
  - Communication failures
  - Arrangements were not kept
- Difficulties:
  - Different time zones
- Successes and good experiences:
  - Finding a good data set
  - Implementation of all four methods
  - Working in a group
  - Easy to find tutorials with keras, pytorch and general information on ANNs
- Organisation:
  - Several weekly meetings
  - Communication via WhatsApp

References

1. Huang, C.-J. Multiple-input deep convolutional neural network model for covid-19 forecasting in china. MedRxiv (2020).
2. Omran. Applying deep learning methods on time-series data for forecasting covid-19 in egypt, kuwait, and saudi arabia. Complexity (2021).
3. Bandyopadhyay, S. K. & Dutta, S. Machine learning approach for confirmation of covid-19 cases: Positive, negative, death and release. medRxiv (2020).
4. Arora, P. Prediction and analysis of covid-19 positive cases using deep learning models: A descriptive case study of india. Chaos, Solitons Fractals 139 (2020).
5. Frausto-Solís, J., Hernández-González, L. J., González-Barbosa, J. J., Sánchez-Hernández, J. P. & Román-Rangel, E. Convolutional neural network–component transformation (cnn–ct) for confirmed covid-19 cases. Math. Comput. Appl. 26, 29 (2021).
6. Wang, L., Lin, Z. Q. & Wong, A. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images.Sci. Reports 10 , 1–12 (2020).
7. Mohimont, L., Chemchem, A., Alin, F., Krajecki, M. & Steffenel, L. A. Convolutional neural networks and temporal cnns for covid-19 forecasting in france. Appl. Intell. 1–26 (2021)

References

8. Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain, cornell aeronautical laboratory. Cornell Aeronaut. Lab. (1958).
9. Géron & Aurélien. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems (O'Reilly Media, 2019).
10. Wang, Y. The influence of the activation function in a convolution neural 419 network model of facial expression recognition. Appl. Sci.(2020).
11. Hochreiter. Lstm can solve hard long time lag problems. Adv. neural information processing systems 473–479 (1997).
12. Cho. Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014).
13. Yuan, X., Li, L. & Wang, Y. Nonlinear dynamic soft sensor modeling with supervised long short-term memory network. IEEE transactions on industrial informatics 16 , 3168–3176 (2019)
14. https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21
15. https://pytorch.org/
16. Raimi Karim, https://towardsdatascience.com/animated-rnn-lstm-and-gru-ef124d06cf45

Thanks for your attention!

Supplementary Slides

Ready to use code

- Ready to use code for production
- API Call with following parameters:
  - Input Data
  - State
  - The Number of future days for which you would like to have output
- Calling API:
  - COVID_INDIA_Prediction(Input_Data,”Delhi”,3)
  - Output: [10500,10600,10700]

Results

CNN model

	State	MAE	RMSE	R2_Score
0	Delhi	0.014204	0.015171	0.946650
1	Kerala	0.015552	0.017611	0.985757
2	Maharashtra	0.005678	0.006385	0.992925
3	Tamil-Nadu	0.028000	0.031074	0.971701
4	Uttar-Pradesh	0.018833	0.020300	0.947357

MLP model

	State	MAE	RMSE	R2_Score
0	Delhi	0.070491	0.072939	-0.233222
1	Kerala	0.053683	0.063735	0.910945
2	Maharashtra	0.023995	0.026345	0.879576
3	Tamil-Nadu	0.068678	0.087206	0.856808
4	Uttar-Pradesh	0.056291	0.062394	0.927715

LSTM model

	State	MAE	RMSE	R2_Score
0	Delhi	0.015091	0.017657	0.991590
1	Kerala	0.022255	0.024993	0.984097
2	Maharashtra	0.006111	0.007037	0.998121
3	Tamil-Nadu	0.029650	0.040859	0.969103
4	Uttar-Pradesh	0.025056	0.033540	0.979112

GRU model

	State	MAE	RMSE	R2_Score
0	Delhi	0.032771	0.034352	0.947925
1	Kerala	0.056093	0.068102	0.898323
2	Maharashtra	0.004072	0.005124	0.998916
3	Tamil-Nadu	0.052893	0.067702	0.915171
4	Uttar-Pradesh	0.072771	0.081398	0.876975