Assessing Covid-19 Situation in Vietnam Using a Data-Driven Epidemiological Compartmental Model

Vo Le Tung
Assessor: Assoc. Prof. Huynh Trung Hieu
Co-Assessor: Assoc. Prof. Nguyen Tuan Duc

Vietnamese - German University
Frankfurt University of Applied Sciences

January 27, 2022

Outline

1. Introduction
2. Literature review
3. Methodologies
4. Results
5. Discussion
6. Conclusion

Outline for Introduction

1 Introduction

The Covid-19 disease

- Health impacts (WHO)
  - 230 million global infections
  - 4.7 million global deaths

The Covid-19 disease

- Health impacts (WHO)
  - 230 million global infections
  - 4.7 million global deaths
- Economic impacts
  - Delayed shipments
  - Increased transaction cost

Why do we need a Covid-19 model?

- Lack of research on the impacts of Non-Pharmaceutical Intervention (NPI) in Vietnam

Why do we need a Covid-19 model?

- Lack of research on the impacts of Non-Pharmaceutical Intervention (NPI) in Vietnam
- Need of a simple tool for assessing the disease current and future progression

Goals

- A model that works with the data availability in Vietnam

Goals

- A model that works with the data availability in Vietnam
- An explainable model that can be used by experts

Goals

- A model that works with the data availability in Vietnam
- An explainable model that can be used by experts
- Assess the effectiveness of mobility data in predicting future cases

Outline for Literature review

2. Literature review
   - Mathematical models
   - Data-driven models
   - Novel compartmental models

Compartmental models

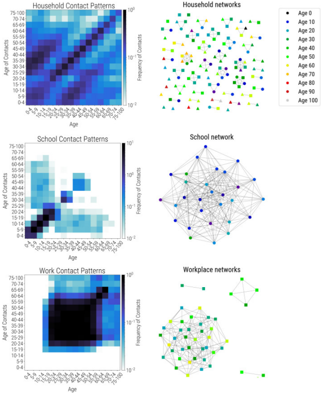
Susceptible-Exposed-Infective-Removed (SEIR) model [1]

$$
\begin{aligned}
S' &= -\beta(N)SI \\
E' &= \beta(N)SI - \kappa E \\
I' &= \kappa E - \alpha I \\
R' &= f\alpha I \\
N' &= -(1-f)\alpha I
\end{aligned}
$$

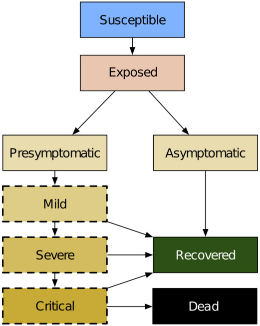
$$\mathrm{S} \xrightarrow{\beta} \mathrm{E} \xrightarrow{\kappa} \mathrm{I} \xrightarrow{\alpha} \mathrm{R}$$

Figure: States graph for the SEIR model

Agent-based models



(a) Transmission networks



(b) States graph

Figure: Covasim model [2]

Pros & cons

Pros

- Explainable
- Based on many years of research
- Easy to understand and implement

Pros & cons

Pros

- Explainable
- Based on many years of research
- Easy to understand and implement

Cons

- Low representational capabilities
- The represented dynamics are stationary
- Unrealistic assumptions about real-world scenarios

Examples

- Autoregressive Integrated Moving Average (ARIMA) models [3]–[5]

Examples

- Autoregressive Integrated Moving Average (ARIMA) models [3]–[5]
- Explainable Artificial Neural Network (ANN) encoder [6]

Examples

- Autoregressive Integrated Moving Average (ARIMA) models [3]–[5]
- Explainable Artificial Neural Network (ANN) encoder [6]
- Recurrent Neural Network (RNN) [7], [8]
    - Long Short Term Memory (LSTM)
    - Bidirectional Long Short Term Memory (Bi-LSTM)
    - Gated Recurrent Unit (GRU)

Pros & cons

Pros

- High prediction accuracy
- Allow for modeling without needing domain knowledge

Pros & cons

Pros

- High prediction accuracy
- Allow for modeling without needing domain knowledge

Cons

- Unexplainable
- Relied on large amount of data
- Inability to capture the true disease dynamics

Incorporating mobility data

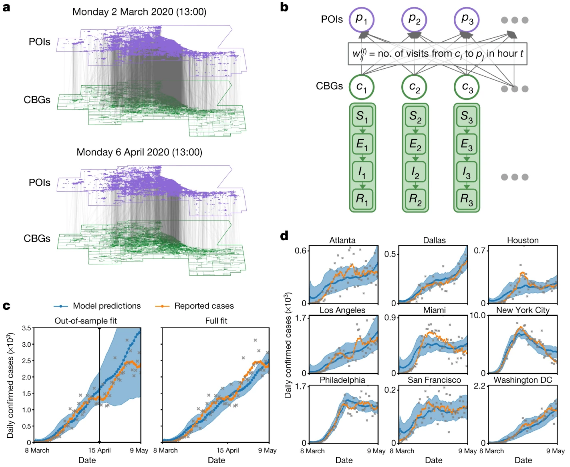
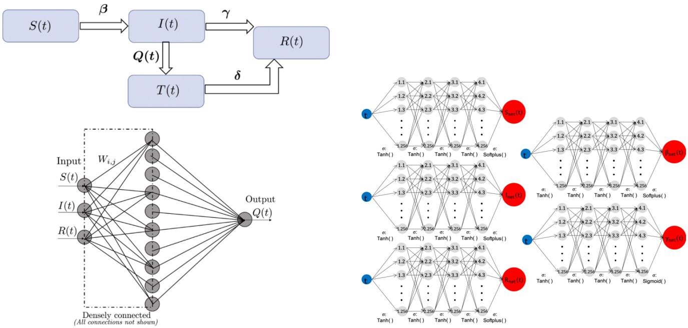


Figure: SEIR model informed with mobility data [9]

Incorporating artificial neural networks



(a) QSIR [10] (b) Time-dependent SIR [11]

Figure: Predict the parameters of compartmental models with Artificial Neural Networks (ANNs)

Outline for Methodologies

3. Methodologies
   - Model definition
   - Datasets
   - Parameters estimation
   - Experiments

Universal differential equation

Universal Differential Equation (UDE) [12] is a neural network architecture based on Neural Ordinary Differential Equation (NeuralODE) [13]

Universal differential equation

UDE [12] is a neural network architecture based on NeuralODE [13]

Generalized NeuralODE

$$u' = U_\theta(u, t)$$

Universal differential equation

UDE [12] is a neural network architecture based on NeuralODE [13]

Generalized NeuralODE

$$u' = U_\theta(u, t)$$

Generalized UDE

$$u' = f(u, t, U_\theta(u, t))$$

System of ODEs

Definition

$$
\begin{aligned}
S' &= -\frac{\beta(t)SI}{N} \\
E' &= \frac{\beta(t)SI}{N} - \gamma E \\
I' &= \gamma E - \lambda I \\
R' &= (1-\alpha(t))\lambda I \\
D' &= \alpha(t)\lambda I \\
N' &= -\alpha\lambda I \\
C' &= -C + \gamma E \\
T' &= \gamma E
\end{aligned}
$$

System of ODEs

Definition

$$
\begin{aligned}
S' &= -\frac{\beta(t)SI}{N} \\
E' &= \frac{\beta(t)SI}{N} - \gamma E \\
I' &= \gamma E - \lambda I \\
R' &= (1-\alpha(t))\lambda I \\
D' &= \alpha(t)\lambda I \\
N' &= -\alpha\lambda I \\
C' &= -C + \gamma E \\
T' &= \gamma E
\end{aligned}
$$

Reproduction number

$$\mathcal{R}^* = \frac{\beta(t)}{\gamma}$$

States graph

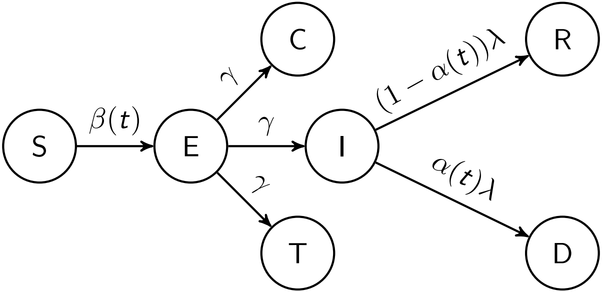


Figure: States graph for the proposed model

System parameters

Box constraints for time-independent parameters

$$\gamma = \gamma_L + (\gamma_U - \gamma_L) * \sigma(\gamma')$$
$$\lambda = \lambda_L + (\lambda_U - \lambda_L) * \sigma(\lambda')$$

System parameters

Box constraints for time-independent parameters

$$\gamma = \gamma_L + (\gamma_U - \gamma_L) * \sigma(\gamma')$$
$$\lambda = \lambda_L + (\lambda_U - \lambda_L) * \sigma(\lambda')$$

Neural networks for time-dependent parameters

$$\beta(t) = \mathcal{NN}_{\theta_1}(\mathcal{F})$$
$$\alpha(t) = \mathcal{NN}_{\theta_2}\left(\frac{t}{t_{\max}}, \frac{I(t-1)}{N(t-1)}, \frac{R(t-1)}{N(t-1)}, \frac{D(t-1)}{N(t-1)}\right)$$

- Hidden layers activation: mish [14]
- Output layer activation $\nu_i = \nu_{i,L} + (\nu_{i,U} - \nu_{i,L}) * \sigma(z_i)$

Neural network input features

1st. set of input features

$$\mathcal{F}_1(t) = \left\{ \frac{t}{t_{\text{max}}}, \frac{S(t-1)}{N(t-1)}, \frac{E(t-1)}{N(t-1)}, \frac{I(t-1)}{N(t-1)} \right\}$$

Neural network input features

1st. set of input features

$$\mathcal{F}_1(t) = \left\{ \frac{t}{t_{\max}}, \frac{S(t-1)}{N(t-1)}, \frac{E(t-1)}{N(t-1)}, \frac{I(t-1)}{N(t-1)} \right\}$$

2nd. set of input features

$$\mathcal{F}_2(t) = \mathcal{F}_1(t) \cup \{\text{MovementRange}(t)\}$$

Neural network input features

1st. set of input features

$$\mathcal{F}_1(t) = \left\{ \frac{t}{t_{\text{max}}}, \frac{S(t-1)}{N(t-1)}, \frac{E(t-1)}{N(t-1)}, \frac{I(t-1)}{N(t-1)} \right\}$$

2nd. set of input features

$$\mathcal{F}_2(t) = \mathcal{F}_1(t) \cup \{\text{MovementRange}(t)\}$$

3rd. set of input features

$$\mathcal{F}_3(t) = \mathcal{F}_2(t) \cup \{\text{SocialProximityToCases}(t)\}$$

Covid-19 cases time series

date	infective	confirmed	recoveries	deaths
1/22/20	0	0	0	0
1/23/20	2	2	0	0
1/24/20	2	2	0	0
...	...	...	...	...

Table: Structure of the processed Covid-19 time series datasets

.

- John Hopkins University (JHU) Covid-19 public datasets
- *VNExpress* Covid-19 public dashboard
- *VnCDC* Covid-19 public dashboard

Facebook's movement range maps dataset

ds	country	polygon	rel. change	stay put ratio
...	...	...	...	...
2021-01-01	VNM	VNM.1.10_1	0.125	0.270
2021-01-02	VNM	VNM.1.10_1	0.052	0.259
2021-01-03	VNM	VNM.1.10_1	0.185	0.269
...	...	...	...	...

Table: Structure of Facebook's Movement Range Maps (MRMs) dataset.

Facebook's social connectedness index

Social Connectedness Index (SCI)

$$\text{SCI}_{i,j} = \frac{\text{FB connections}_{i,j}}{\text{FB users}_i * \text{FB users}_j},$$

Facebook's social connectedness index

Social Connectedness Index (SCI)

$$\text{SCI}_{i,j} = \frac{\text{FB connections}_{i,j}}{\text{FB users}_i * \text{FB users}_j},$$

Social Proximity to Cases (SPC) index [15]

$$\text{SPC}_{i,t} = \sum_{j}^{C} \text{Cases per 10k}_{j,t} \frac{\text{SCI}_{i,j}}{\sum_h^C \text{SCI}_{i,h}},$$

Population data

ID_1	NAME_1	AVGPOPULATION
3	Ha Noi	8.2466e6
62	Vinh Phuc	1.1712e6
...	...	...
1001.0	Autauga, Alabama, US	55869
1003.0	Baldwin, Alabama, US	223234
...	...	...

Table: Structure of the processed average population datasets

.

- Vietnam General Statistics Office (GSO)
- JHU Covid-19 datasets

Loss function

Regularized Mean Squared Error (MSE) with scaled outputs

$$\mathcal{L}(\hat{y}, y) = \frac{1}{T} \sum_{i=1}^{N} \sum_{t=0}^{T-1} \left[e^{\zeta t} \left(\frac{\hat{y}_{i,t} - y_{i,t}}{max(y_i) - min(y_i)} \right)^2 \right] + \frac{\lambda}{2T} (\|\theta_1\|_2^2 + \|\theta_2\|_2^2)$$

Training process

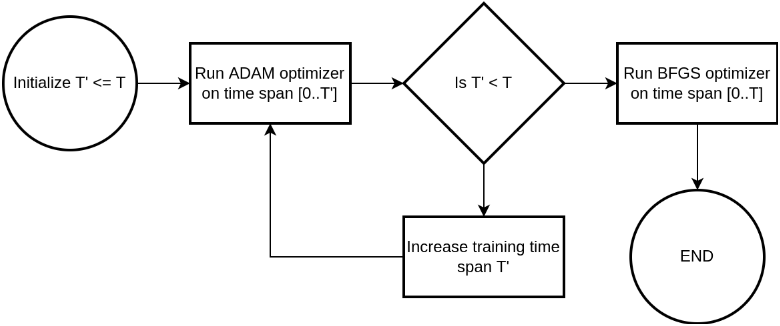


Figure: Model training procedure

Data preprocessing

- Applied 7-day moving average to all datasets
- Applied min-max scaling on the Movement Range Map (MRM) dataset and Social Proximity to Cases (SPC) index

Data preprocessing

- Applied 7-day moving average to all datasets
- Applied min-max scaling on the Movement Range Map (MRM) dataset and Social Proximity to Cases (SPC) index
- Training period: 48 days
  - Vietnam: First date when the total confirmed cases passed 500
  - United States: 1st July 2021

Data preprocessing

- Applied 7-day moving average to all datasets
- Applied min-max scaling on the Movement Range Map (MRM) dataset and Social Proximity to Cases (SPC) index
- Training period: 48 days
  - Vietnam: First date when the total confirmed cases passed 500
  - United States: 1st July 2021
- Testing period: 28 days

Differential equation solver

- Tsit5 solver (Tsitouras Runge-Kutta 5/4 method [16])
- InterpolatingAdjoint technique for approximating gradients in NeuralODE [17]

Initial conditions

Location	$S(0)$	$E(0)$	$I(0)$	$R(0)$
Vietnam	9.75e7	25	5	2817
Ho Chi Minh city	9.22e6	173.5	34.7	360.1
Binh Duong	2.58e6	206.4	41.2	314.7
Dong Nai	3.17e6	295	59	194.8
Long An	1.71e6	217.8	43.5	300.5
United States	2.99e8	71890	14378	3.31e7
Los Angeles, California	8.78e6	2500	500	1.22e6
Cook, Illinois	4.59e6	615	123	546508
Harris, Texas	4.30e6	930	186	396430
Maricopa, Arizona	3.92e6	1325	265	549899

Table: Initial conditions for the system of Ordinary Differential Equations (ODEs)

Initial parameters

Parameter	Value	Lower bound	Upper bound
β	*N/A*	0.05	1.67
γ	$1/4$	$1/4$	$1/4$
λ	$1/14$	$1/14$	$1/14$
α	*N/A*	0.005	0.05
θ_1	*glorot_normal*	*N/A*	*N/A*
θ_2	*glorot_normal*	*N/A*	*N/A*

Table: Initial parameters for the system of ODEs

Outline for Results

4 Results

- Model's outputs for Vietnam and the United States
- Model's outputs for counties in the United States
- Model's outputs for provinces in Vietnam

Evaluation metric

Mean absolute percentage error

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{\hat{y}_i - y_i}{y_i} \right|$$

MAPE: Country-level data

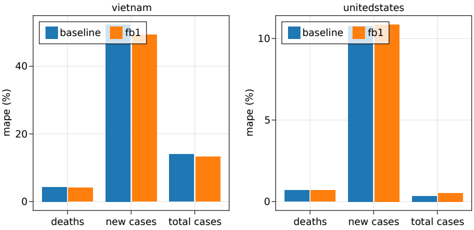


Figure: MAPE for 28-day forecast horizon

Forecasts: Country-level data

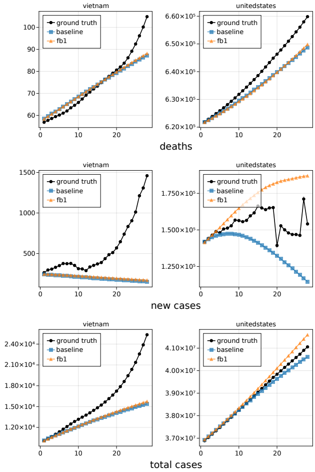


Figure: Forecasts made by different versions of the model

MAPE: Counties in the United States

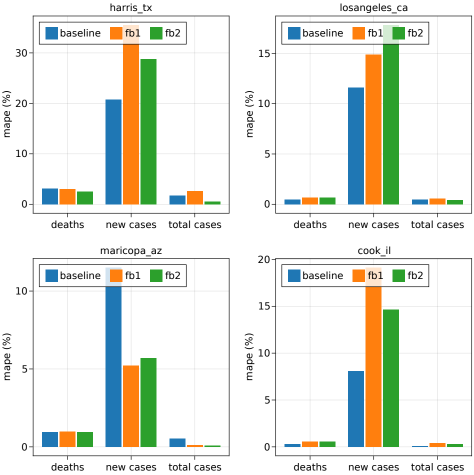


Figure: MAPE for 28-day forecast horizon

Forecasts: Counties in the United States

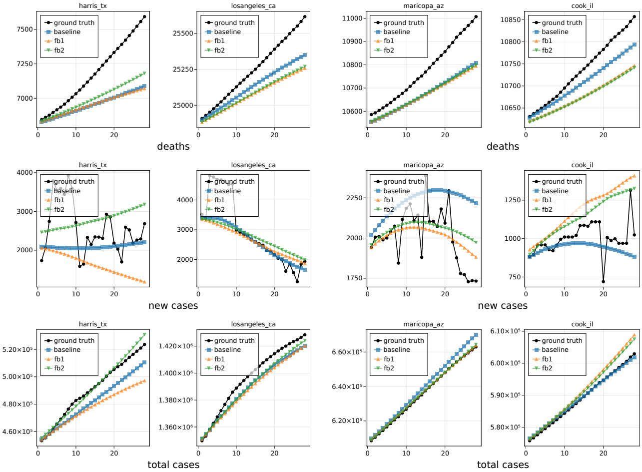


Figure: Forecasts made by different versions of the model

MAPE: Provinces in Vietnam

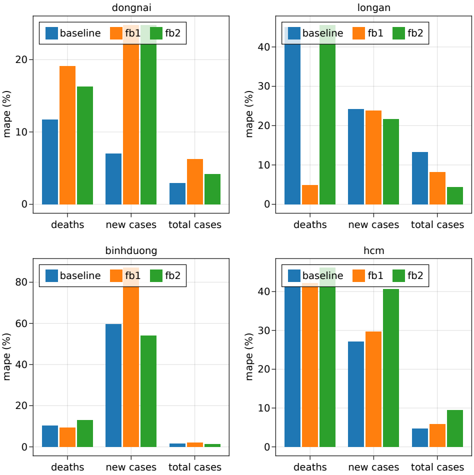


Figure: MAPE for 28-day forecast horizon

Forecasts: Provinces in Vietnam

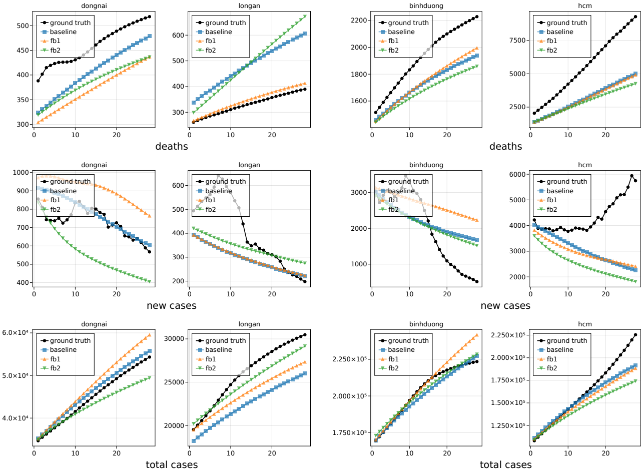


Figure: Forecasts made by different versions of the model

Outline for Discussion

5 Discussion
- Model's convergence and generalizability
- Model's limitations

About the results

- Could capture the disease trends

About the results

- Could capture the disease trends
- Accuracy get lower as we extrapolated further into the future

About the results

- Could capture the disease trends
- Accuracy get lower as we extrapolated further into the future
- Bad fit and low forecast accuracy when applied to data with high fluctuation

About the results

- Could capture the disease trends
- Accuracy get lower as we extrapolated further into the future
- Bad fit and low forecast accuracy when applied to data with high fluctuation
- No improvement in the forecast accuracy when mobility data was incorporated

Model's convergence and generalizability

- The model could get stuck in bad local minima, due to
    - ODEs *stiffness*
    - High frequency changes

Model's convergence and generalizability

- The model could get stuck in bad local minima, due to
  - ODEs *stiffness*
  - High frequency changes
- The model was separately trained at each location and only capable of making forecast for the location that it was trained with

Model's convergence and generalizability

- The model could get stuck in bad local minima, due to
    - ODEs *stiffness*
    - High frequency changes
- The model was separately trained at each location and only capable of making forecast for the location that it was trained with
- Forecast can only be made for a short period after the training period

Model's limitations

- Issues with ground truth data

Model's limitations

- Issues with ground truth data
- Inability to capture rapid trend changes

Model's limitations

- Issues with ground truth data
- Inability to capture rapid trend changes
- Bad local minima

Model's limitations

- Issues with ground truth data
- Inability to capture rapid trend changes
- Bad local minima
- Compartmental model bias

Model's limitations

- Issues with ground truth data
- Inability to capture rapid trend changes
- Bad local minima
- Compartmental model bias
- Covariates bias

Model's limitations

- Issues with ground truth data
- Inability to capture rapid trend changes
- Bad local minima
- Compartmental model bias
- Covariates bias
- The changing dynamics of Covid-19

Model's limitations

- Issues with ground truth data
- Inability to capture rapid trend changes
- Bad local minima
- Compartmental model bias
- Covariates bias
- The changing dynamics of Covid-19
- Neural network interpretability

Outline for Conclusion

6 Conclusion

What has been done?

Goals: An explainable Covid-19 model that can work with the low data availability in Vietnam

What has been done?

Goals: An explainable Covid-19 model that can work with the low data availability in Vietnam

- Collect and process Covid-19 related data for various locations

What has been done?

Goals: An explainable Covid-19 model that can work with the low data availability in Vietnam

- Collect and process Covid-19 related data for various locations
- Implement a model an explainable Covid-19 model using UDE
  - Progression of all compartments can be given by the model
  - Metric about the disease, e.g. $\mathcal{R}_0$, can be derived

What has been done?

Goals: An explainable Covid-19 model that can work with the low data availability in Vietnam

- Collect and process Covid-19 related data for various locations
- Implement a model an explainable Covid-19 model using UDE
  - Progression of all compartments can be given by the model
  - Metric about the disease, e.g. $\mathcal{R}_0$, can be derived
- Add mobility data to the model and show that ineffectiveness of the approach

Future improvements

- Increase the model complexity with additional compartments and parameters

Future improvements

- Increase the model complexity with additional compartments and parameters
- Include additional time-varying covariates

Future improvements

- Increase the model complexity with additional compartments and parameters
- Include additional time-varying covariates
- Implement better methods and algorithms for training UDE

Future improvements

- Increase the model complexity with additional compartments and parameters
- Include additional time-varying covariates
- Implement better methods and algorithms for training UDE
- Use methods such as Sparse Identification of Nonlinear Dynamical System (SINDy) to find a governing equation and improve the model interpretability

References I

[1] F. Brauer, "Compartmental Models in Epidemiology," in *Mathematical Epidemiology*, ser. Lecture Notes in Mathematics, F. Brauer, P. van den Driessche, and J. Wu, Eds., red. by J. .-. Morel, F. Takens, and B. Teissier, vol. 1945, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 19–79.

[2] C. C. Kerr, R. M. Stuart, D. Mistry, *et al.*, "Covasim: An agent-based model of COVID-19 dynamics and interventions," *PLOS Computational Biology*, vol. 17, no. 7, e1009149, Jul. 26, 2021.

[3] Z. Ceylan, "Estimation of COVID-19 prevalence in Italy, Spain, and France," *Science of The Total Environment*, vol. 729, p. 138 817, Aug. 10, 2020.

References II

[4] R. K. Singh, M. Rani, A. S. Bhagavathula, *et al.*, "Prediction of the COVID-19 Pandemic for the Top 15 Affected Countries: Advanced Autoregressive Integrated Moving Average (ARIMA) Model," *JMIR Public Health and Surveillance*, vol. 6, no. 2, e19115, May 13, 2020.

[5] M. H. D. M. Ribeiro, R. G. da Silva, V. C. Mariani, and L. d. S. Coelho, "Short-term forecasting COVID-19 cumulative confirmed cases: Perspectives for Brazil," *Chaos, Solitons & Fractals*, vol. 135, p. 109 853, Jun. 1, 2020.

[6] A. Ramchandani, C. Fan, and A. Mostafavi, "DeepCOVIDNet: An Interpretable Deep Learning Model for Predictive Surveillance of COVID-19 Using Heterogeneous Features and Their Interactions," *IEEE Access*, vol. 8, pp. 159 915–159 930, 2020.

References III

[7] V. K. R. Chimmula and L. Zhang, "Time series forecasting of COVID-19 transmission in Canada using LSTM networks," *Chaos, Solitons, and Fractals*, vol. 135, p. 109 864, Jun. 2020.

[8] F. Shahid, A. Zameer, and M. Muneeb, "Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM," *Chaos, Solitons, and Fractals*, vol. 140, p. 110 212, Nov. 2020.

[9] S. Chang, E. Pierson, P. W. Koh, *et al.*, "Mobility network models of COVID-19 explain inequities and inform reopening," *Nature*, vol. 589, no. 7840, pp. 82–87, 7840 Jan. 2021.

References IV

[10] R. Dandekar, C. Rackauckas, and G. Barbastathis, “A Machine Learning-Aided Global Diagnostic and Comparative Tool to Assess Effect of Quarantine Control in COVID-19 Spread,” *Patterns*, vol. 1, no. 9, p. 100 145, Dec. 11, 2020.

[11] S. Y. Jung, H. Jo, H. Son, and H. J. Hwang, “Real-World Implications of a Rapidly Responsive COVID-19 Spread Model with Time-Dependent Parameters via Deep Learning: Model Development and Validation,” *Journal of Medical Internet Research*, vol. 22, no. 9, e19907, Sep. 9, 2020.

[12] C. Rackauckas, Y. Ma, J. Martensen, *et al.*, “Universal Differential Equations for Scientific Machine Learning,” (Aug. 6, 2020), [Online]. Available: http://arxiv.org/abs/2001.04385 (visited on 09/11/2021).

References V

[13] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural Ordinary Differential Equations," (Dec. 13, 2019), [Online]. Available: http://arxiv.org/abs/1806.07366 (visited on 09/26/2021).

[14] D. Misra, "Mish: A Self Regularized Non-Monotonic Activation Function," (Aug. 13, 2020), [Online]. Available: http://arxiv.org/abs/1908.08681 (visited on 12/06/2021).

[15] T. Kuchler, D. Russel, and J. Stroebel, "The Geographic Spread of COVID-19 Correlates with the Structure of Social Networks as Measured by Facebook," National Bureau of Economic Research, Working Paper 26990, Apr. 2020.

References VI

[16] C. Tsitouras, "RungeKutta pairs of order 5(4) satisfying only the first column simplifying assumption," *Computers & Mathematics with Applications*, vol. 62, no. 2, pp. 770–775, Jul. 2011.

[17] T. Daulbaev, A. Katrutsa, L. Markeeva, J. Gusak, A. Cichocki, and I. Oseledets, "Interpolation Technique to Speed Up Gradients Propagation in Neural ODEs," (Oct. 30, 2020), [Online]. Available: http://arxiv.org/abs/2003.05271 (visited on 01/26/2022).

[18] S. Zhao and H. Chen, "Modeling the epidemic dynamics and control of COVID-19 outbreak in China," *Quantitative Biology*, vol. 8, no. 1, pp. 11–19, Mar. 2020.

References VII

[19] S. He, Y. Peng, and K. Sun, "SEIR modeling of the COVID-19 and its dynamics," *Nonlinear Dynamics*, vol. 101, no. 3, pp. 1667–1680, Aug. 1, 2020.

[20] F. Ndaïrou, I. Area, J. J. Nieto, and D. F. Torres, "Mathematical modeling of COVID-19 transmission dynamics with a case study of Wuhan," *Chaos, Solitons, and Fractals*, vol. 135, p. 109 846, Jun. 2020.

[21] S. B. Bastos and D. O. Cajueiro, "Modeling and forecasting the early evolution of the Covid-19 pandemic in Brazil," *Scientific Reports*, vol. 10, no. 1, p. 19 457, Dec. 2020.

[22] K. Sarkar, S. Khajanchi, and J. J. Nieto, "Modeling and forecasting the COVID-19 pandemic in India," *Chaos, Solitons, and Fractals*, vol. 139, p. 110 049, Oct. 2020.

References VIII

[23] G. Cybenkot, "Approximation by superpositions of a sigmoidal function,", p. 12,

[24] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.

[25] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, Jan. 1989.

[26] Y. Guo, X. Cao, L. Bainian, and M. Gao, "Solving Partial Differential Equations Using Deep Learning and Physical Constraints," *Applied Sciences*, vol. 10, p. 5917, Aug. 26, 2020.

Outline for Mathematical models

7 Mathematical models

Improvements for Covid-19

Changes made to classical models for infectious disease [18]–[22]:

Improvements for Covid-19

Changes made to classical models for infectious disease [18]–[22]:

- Inclusion of compartments that represent government interventions

Improvements for Covid-19

Changes made to classical models for infectious disease [18]–[22]:

- Inclusion of compartments that represent government interventions
- Inclusion of compartments that specifically represent the behavior of SARS-NCOV-2

Improvements for Covid-19

Changes made to classical models for infectious disease [18]–[22]:

- Inclusion of compartments that represent government interventions
- Inclusion of compartments that specifically represent the behavior of SARS-NCOV-2
- Separation of the infectious compartment into multiple compartments representing the severity of the patients

Outline for MLPs

8 MLPs

Multi-layer perceptron: Graph representation

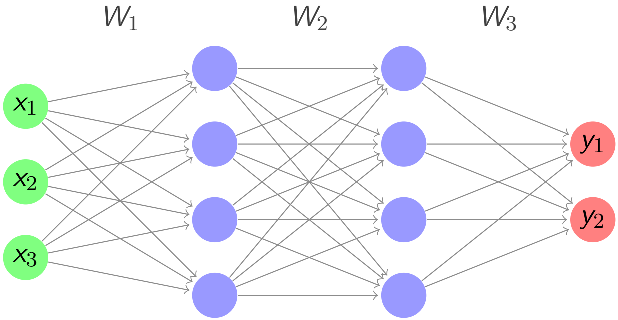


Figure: A Multi-Layer Perceptron (MLP) with four layers

Multi-layer perceptron: Mathematical representation

Definition

$$g(X) = \phi_n(W_n\phi_{n-1}(\cdots(W_2\phi_1(W_1X + b_1) + b_2) + \cdots) + b_n)$$

Multi-layer perceptron: Mathematical representation

Definition

$$g(X) = \phi_n(W_n \phi_{n-1}(\cdots(W_2 \phi_1(W_1 X + b_1) + b_2) + \cdots) + b_n)$$

Theorem

Given appropriate weights, ANNs can approximate any arbitrary function $f : \mathbb{R}^M \to \mathbb{R}^N$ [23]–[25]

Training neural networks: Back-propagation

MSE

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(g(X_i) - Y_i)^2$$

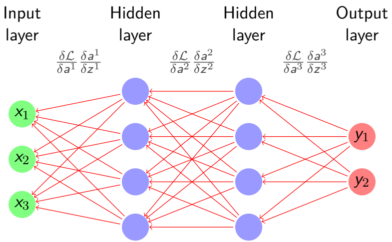


Figure: Graph representation of the back-propagation algorithm

Outline for PINNs

9 PINNs

Physics-informed neural networks

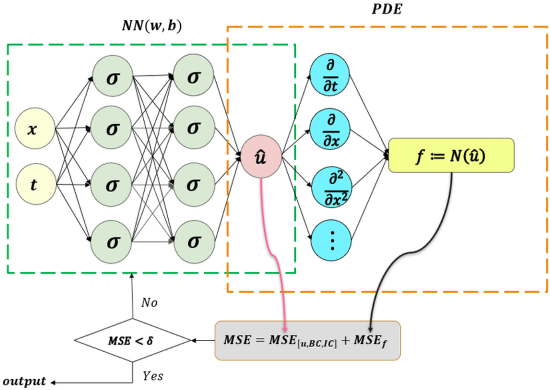


Figure: The schematic of Physics-Informed Neural Networks (PINNs) for solving Partial Differential Equations (PDEs) [26].

Outline for NeuralODEs

10 NeuralODEs

Neural ordinary differential equations: Idea

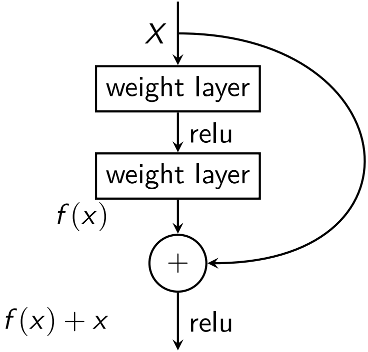


Figure: Skip connection

Neural ordinary differential equations: Idea

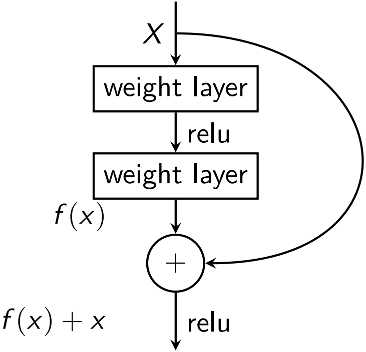


Figure: Skip connection

Observation

$$h_{t+1} = h_t + f(h_t, \theta_t)$$
$$\Leftrightarrow \frac{dh(t)}{dt} = f(h(t), t, \theta)$$

Neural ordinary differential equations: Formulation

Neural Ordinary Differential Equation (NeuralODE) output

$$z(t_1) = z(t_0) + \int_{t_0}^{t_1} f(z(t), t, \theta) dt$$

Neural ordinary differential equations: Formulation

Neural Ordinary Differential Equation (NeuralODE) output

$$z(t_1) = z(t_0) + \int_{t_0}^{t_1} f(z(t), t, \theta) dt$$

- Memory efficiency

Neural ordinary differential equations: Formulation

Neural Ordinary Differential Equation (NeuralODE) output

$$z(t_1) = z(t_0) + \int_{t_0}^{t_1} f(z(t), t, \theta) dt$$

- Memory efficiency
- Adaptive computation

Neural ordinary differential equations: Formulation

Neural Ordinary Differential Equation (NeuralODE) output

$$z(t_1) = z(t_0) + \int_{t_0}^{t_1} f(z(t), t, \theta)dt$$

- Memory efficiency
- Adaptive computation
- Scalable and invertible normalizing flows

Neural ordinary differential equations: Formulation

Neural Ordinary Differential Equation (NeuralODE) output

$$z(t_1) = z(t_0) + \int_{t_0}^{t_1} f(z(t), t, \theta) dt$$

- Memory efficiency
- Adaptive computation
- Scalable and invertible normalizing flows
- Continuous time series models

Neural ordinary differential equations: Outputs

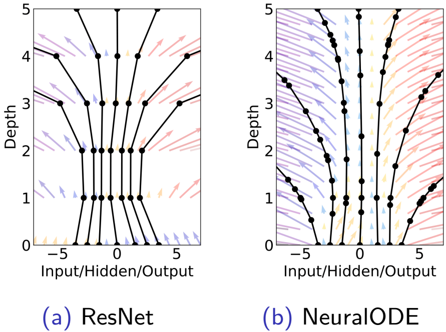


Figure: Comparison between ResNet's discrete state transformations and NeuralODE continuous state transformations

Neural ordinary differential equations: Gradients

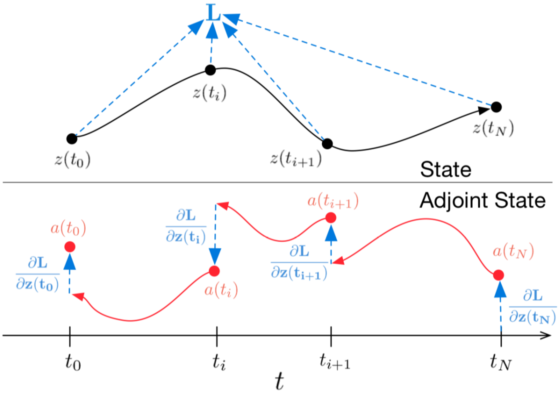


Figure: Reverse-mode differentiation of an ODE solution. [13]

Outline for Software and hardware

11 Software and hardware

Software and hardware

Julia programming language

- *DifferentialEquations* package
- *DiffEqFlux* package

Linux systems

- Google Cloud Compute *n2-standard-8* instance
- Personal laptop with 2-core *Intel(R) Core(TM) i5-4260U* CPU 1.40GHz, and 4Gb of memory.

Julia example

```julia
function SEIRD!(du, u, p, t)
    @inbounds begin
        S, E, I, _, _, N, C, _ = u
        β, γ, λ, α = p
        du[1] = -β * S * I / N
        du[2] = β * S * I / N - γ * E
        du[3] = γ * E - λ * I
        du[4] = (1 - α) * λ * I
        du[5] = α * λ * I
        du[6] = -α * λ * I
        du[7] = -C + γ * E
        du[8] = γ * E
    end
    return nothing
end
```

Figure: Implementation of the system of ODEs in Julia

Outline for Hyperparameters

12 Hyperparameters

Hyperparameters

- Initial time span of 4 days
- Time span increment of 4 days

Hyperparameters

- Initial time span of 4 days
- Time span increment of 4 days
- Time weighting parameter $\zeta = -0.001$

Hyperparameters

- Initial time span of 4 days
- Time span increment of 4 days
- Time weighting parameter $\zeta = -0.001$
- ADAM optimizer
  - Learning rate 0.05
    - decay rate 0.5
    - decay step 1000
    - decay limit 0.00001
  - 1000 iterations on each time span

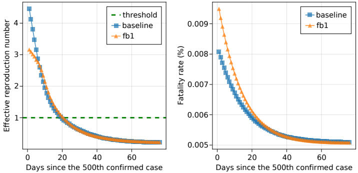
Hyperparameters

- Initial time span of 4 days
- Time span increment of 4 days
- Time weighting parameter $\zeta = -0.001$
- ADAM optimizer
  - Learning rate 0.05
    - decay rate 0.5
    - decay step 1000
    - decay limit 0.00001
  - 1000 iterations on each time span
- BFGS optimizer
  - *Initial stepnorm* 0.01
  - 1000 iterations on the full time span

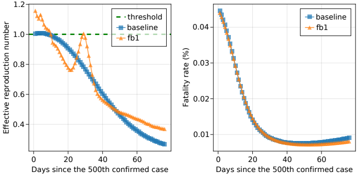
Outline for R_0 and fatality rate

13 R_0 and fatality rate

R_0 and fatality rate: Country-level data



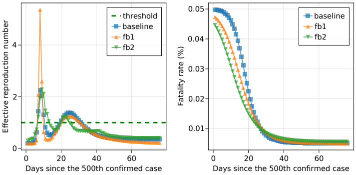
(a) Viet nam



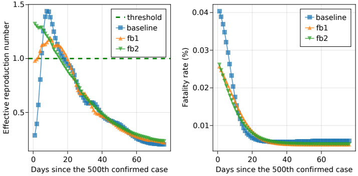
(b) United States

Figure: Disease metrics learned by different versions of the model

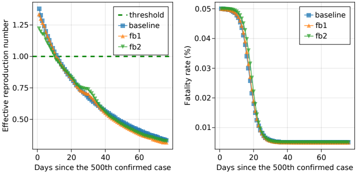
R_0 and fatality rate: Counties in the United States



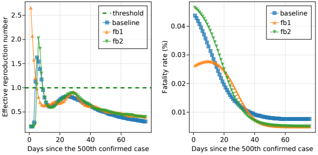
(a) Harris, TX



(b) Los Angeles, CA



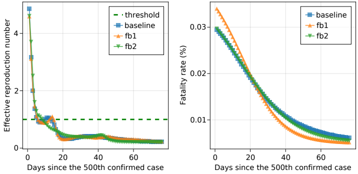
(c) Maricopa, AZ



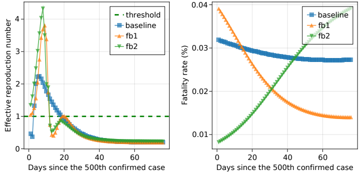
(d) Cook, IL

Figure: Disease metrics learned by different versions of the model

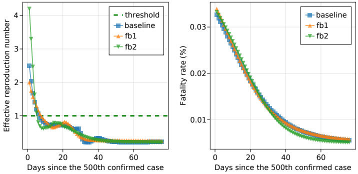
R_0 and fatality rate: Provinces in Vietnam



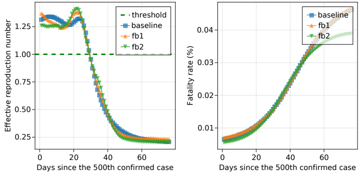
(a) Dong Nai



(b) Long An



(c) Binh Duong



(d) Ho Chi Minh city

Figure: Disease metrics learned by different versions of the model