Forecast of outbreaks and the optimal flow between populations: a scientific machine learning approach .

Matías Núñez
CONICET
Pandemov
Bariloche, Río Negro, Argentina
matias.nunez2@gmail.com

Florencia Grinblat
Pandemov
CABA, Argentina
florencia.grinblat@gmail.com

ABSTRACT

ΔI

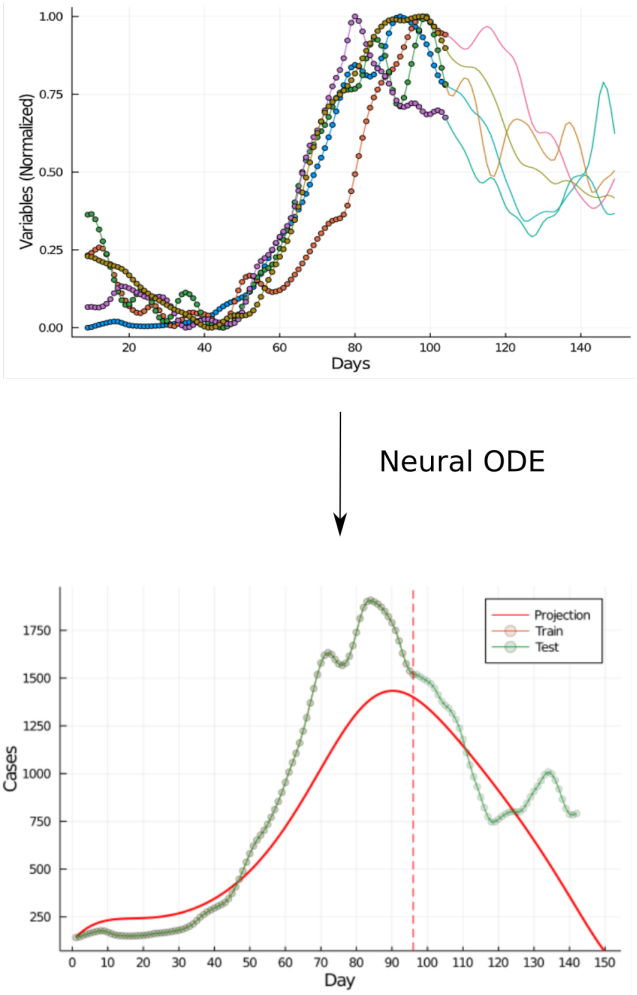
Humanity has been shaped throughout history by the outbreak of infectious diseases. The oldest way to fight against them has been social isolation, quarantine and soap. Currently, in the middle of the COVID 19 pandemic, isolation and restriction on the mobility of people continues to be a method used in various regions of the globe. Specially in countries with precarious health system which are easily overwhelmed. As the acquisition of data and information increases, so do the prevention systems We propose an approach based on ideas of scientific machine learning using neural differential equations (neural ODE) ([**?**]). This is a relatively new kind of network and among all the possible family of neural networks the reason for its selection is twofold. On the one hand we show that they serve as a detector of outbreaks and future dynamics, since due to their characteristics, they are capable of learning the dynamics of the system using relatively little data, and thus make accurate predictions. And on the other, once they have already learned the dynamics, the can serve as models to calculate the optimal flow of people between populations in such a way that the future of positive cases remain within a pre-established level. Among closed populations (states, provinces, countries, etc), which do not allow the entry or exit of people, this kind of approach can help to alleviate the measures of isolation between populations, and thus allow the movement of people and therefore activate stagnant economies.

NOTE: we heard about this challenge two weeks ago, thus we did not have much time. this is an incomplete draft, and a complete text can be found in our blog https://pandemov.blogspot.com/ and github https://github.com/pandemov/challenge .

1 APPROACH

We chose different combinations of variables for an specific location, splited their time series in two, and used their earlier time values to infer the latest ones, with the goal of predicting the number of future infected cases. (ACA why?) The way we approached this was thru the use of a data driven function approximator which is optimized to express the rate of temporal variation of the chosen set of variables . Such an expression, which carries information of the data manifold from the earlier times, is integrated in time for obtaining a possible future time solution trajectory for the set of variables. It is expected that this prediction will deteriorate in time, the further away from the basin manifold, nonetheless there will a time frame of useful information.

We show as a proof of concept, that this newly gained knowledge can be used to correlate the patterns of mobility, flow of people,



The Neural ODE learns from a set of training data correspondent to different chosen variables (above) which belong to a particular region (the state of SC in this case) and tries to learn the dynamics of the system by finding the ordinary differential equation that best describes the data. The learnt solution against the data for Active Cases in SC is shown below. The variables used, that best describe the dynamics are Hospital Admissions, Symptoms, Symptoms in Community, COVID Searches on Google, Doctor Visits and Cases.

Figure 1

between different populations, and their impact on the infected cases .

As a function approximator we used neural ordinary differential equations (neural ODE). A neural ODE learns the underlying dynamics present in the data, by considering a continuous setting and assuming that the change is governed by an ODE

$$\frac{\partial y}{\partial t} = f(y;\theta) \tag{1}$$

to be related through some function $y_i = f(t_i;\theta)$ where θ are learnable parameters. The goal is to learn the underlying dynamics of change. It is a neural network inside an ordinary differential equation . The "forward pass" through a neural ODE is equivalent to solving an initial value problem, where $y(t_0)$ is the input features and we replace hand-crafted equations with a neural network. A single forward pass gives us an entire trajectory. If the dynamics do not change abruptly this has very powerful generalisation capabilities. In contrast , other networks like Resnets (RNNs), for each forward pass through the model it only gives a single prediction in time. To be specific, the forward pass consist of inputting $y(t_0)$ and solve the differential equation forward in time to solve

$$y(t) = y(t_0) + \int_{t_0}^{t} f^*(y(t))\partial t \tag{2}$$

where we use a neural network to model f^* In our case the $y(t)$ is a vector with the values of the set of chosen variables at time t. The parameters of the neural ODE are learned from the data (see Fig. 1) The learning process was performed by minimizing the following loss function

$$L(\theta) = \sum_i (y(t_i) - y_{data}(t_i))^2 \tag{3}$$

in function of the neuronal net parameters θ and where $y_{data}(t_i)$ represent the t data set as a vector which components are values from the chosen set of data variables at time t_i and $y(t_i)$ is the solution of the ordinary differential equation given by Eq. 2.

For optimizing the parameters of the neuronal net we used mini batches and the back propagation thru the ODE solver was done using the ad joint sensitivity method (Ref).

Once the loss function is minimized, and an optimal solution of the neural ODE is found it is extrapolated for assessing its generalization capabilities by comparing with the test data set.

We used the set of tools available in the Julia library DiffEqFlux (https://github.com/SciML/DiffEqFlux.jl).

We make a proof of concept of the use of this newly available information for

les.

Abstract (Briefly describe the background and overall purpose of the analytic approach) (3,500 characters max) * Please describe the background of your analytical approach and use of data set(s) (3,500 characters max)*

2 METHODS

After a data pre processing (We pre-processed the data using Data Science techniques with Pandas, Numpy and csaps written in Python . We used the data in the following link https://cmu-delphi.github.io/delphi-epidata/api/covidcast.html correspondant to the U.S. dataset(s) An

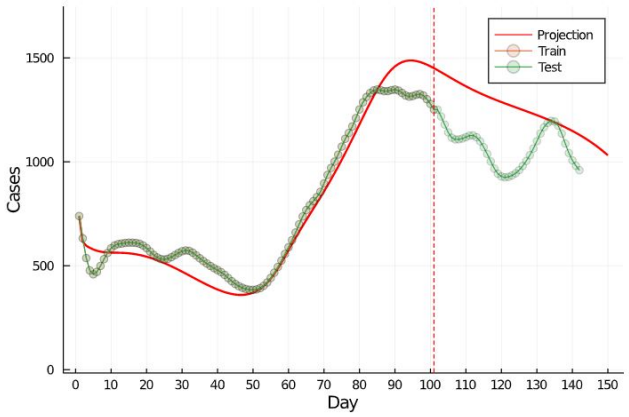


Figure 2: Neural ordinary differential equation (nODE) prediction for the active cases in the state of Ohio . The neuronal net is able to predict the next peak of cases and the average dynamics for almost forty days. The vertical line delimits the data set used to train the neuronal network and from the test set. The solid line is the solution of the neural ODE, which learned the dynamics of the outbreak. The variables used for this forecast are Cases, Hospital Admission, COVID-Like Symptoms, COVID-Like Symptoms in Community and COVID-Related Doctor Visits [?]

example of the smoothing treatment performed to raw data before being used as inputs to the neural networks can be found in our github-link.(https://github.com/pandemov/challenge)) the data.

using the structure of the earlier one.

, inspected their respec time series from a starting time on. from the available variables

We used the neural ODE for forecasting future cases using previous data. The variables we used were mainly .

3 RESULTS

3.1 Relevant Figures and Graphs

UPLOAD graphs, charts, tables, etc. that visualize the findings of your analytical approach **??**.

3.2 LINK to Github for your analytical approach

https://github.com/pandemov/challenge

4 DISCUSSION

Discussion and implications of findings (3,500 characters max)* UPLOAD a presentation (10 slides max) summarizing the analytic approach, findings, and the relevance of your approach to the COVID-19 disease control policy and practice

5 LIMITATIONS AND FURTHER WORK

Describe the limitations of the analytic approaches and discuss intended further work (3,500 characters max)

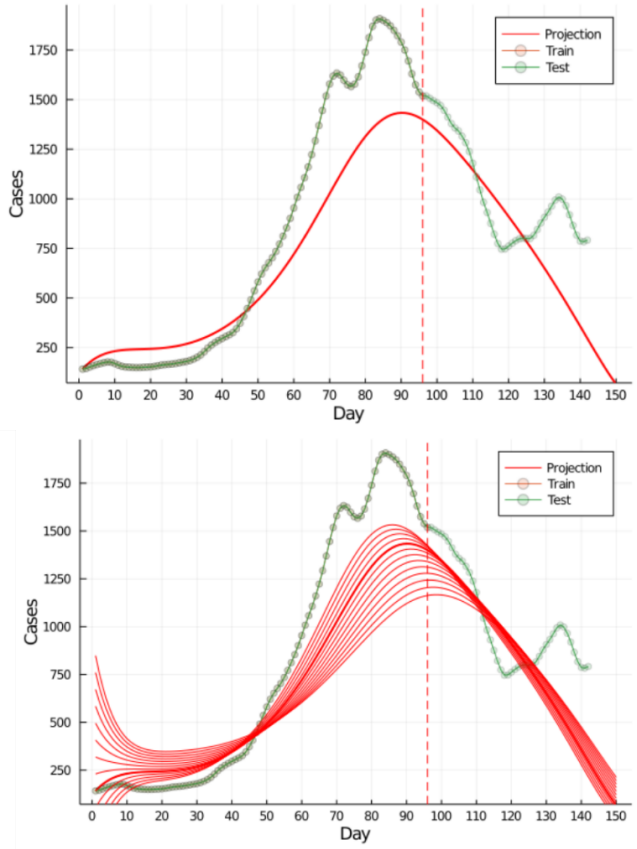


Figure 4: Once the dynamics is learnt by the neural ODE, its solution can predict the projection for different number of initial cases. This can be useful for estimating the error on the forecast, and also for inferring the effect of changes in the number of cases.

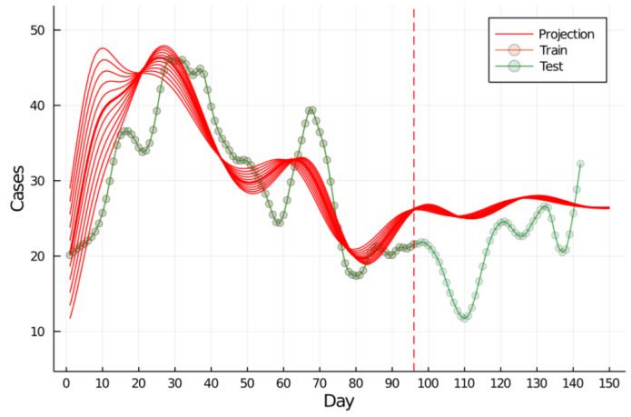


Figure 5: Different solutions for different initial number of infected cases for the state of Main.

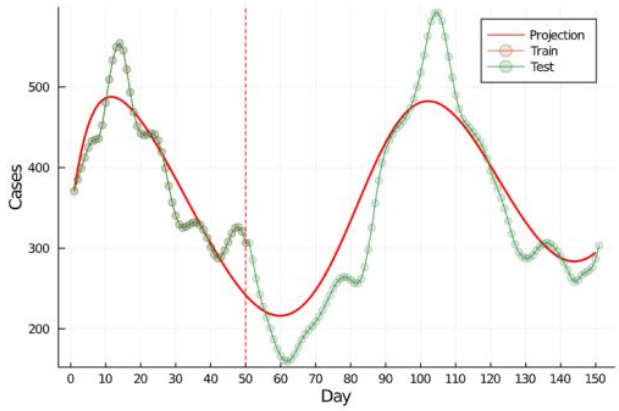


Figure 6: The trainning data set used here (CO state) finished before another uprising of infected cases. Nonetheless the neural ODE is able to detect the outbreak and even accurately pinpoint almost 60 days later the date of the outbreak peak.

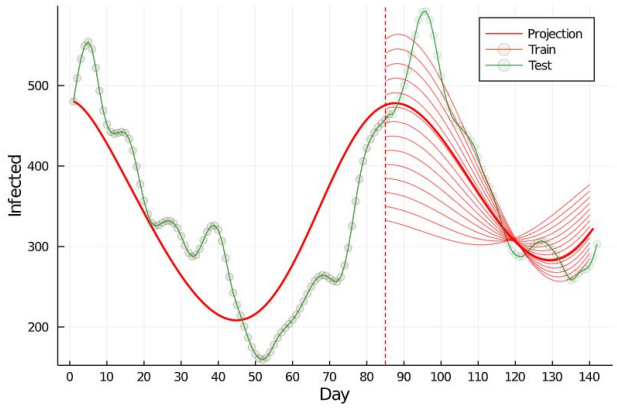
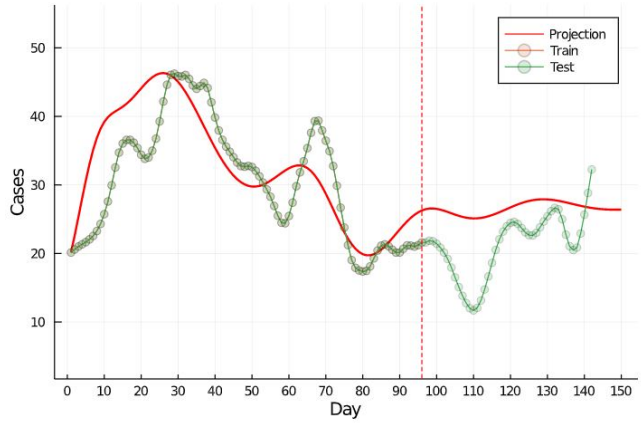


Figure 7: The trained neural ODE can extrapolate different possible scenarios. A change in the number of infected cases can be for instance due to the flow of people traveling in or out of the location given an overall perturbation in the number of infected. This change and its consequences can be estimated by looking at the perturbed solution. This opens the possibility, for locations with strict close borders to be able to predict the effect of the flow of people on the curve of infected.



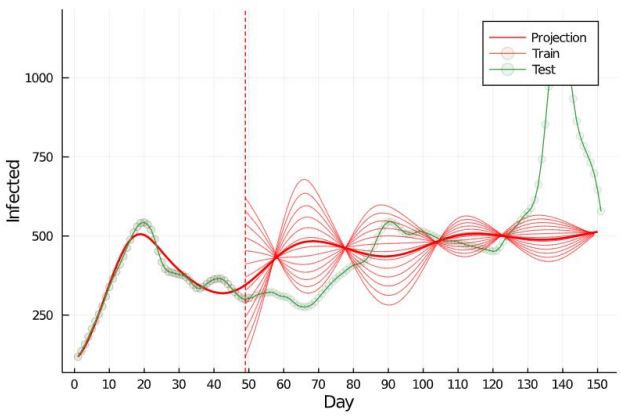


Figure 8: Possible future scenarios for the infected in the state of IA if at the day marked by the vertical line there is a change in number of infected. The different solutions correspond to different initial conditions of the trained neural ODE using the training data.

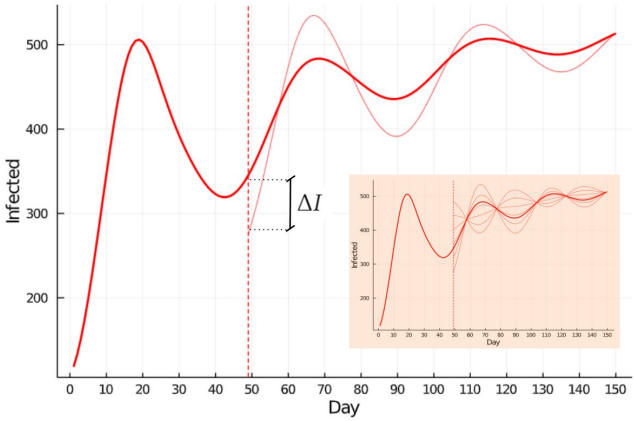


Figure 9: A change in the number of infected ΔI can arise due to the inflow or outflow of people in the region considered. The fact of having a mathematical model in the form of a differential equation allows to calculate and estimate of the change in the infected curve in the future simply by setting a different initial condition and solving the neural ODE (thin line). Morover, one could see the landscape with multiple ΔI (see inset).

ACKNOWLEDGMENTS

This work was supported by us. We also thank the Python and Julia developers for contributing to Data Science, Machine Learning and Artificial Intelligence.

REFERENCES