Latent Variables and the Learnability of Neural Networks

Nicholas Cich
ncich3@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia, USA

Ryan Bauer
rbauer32@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia, USA

Asa Harbin
aharbin6@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia, USA

1 INTRODUCTION

In general, using neural networks to learn the parameters of an ordinary differential equation (ODE) system is quite difficult. Though neural networks can approximate parameter values in such a setting, even small amounts of noise can lead to significant errors that result in poor system forecasting. This issue is further amplified in an epidemiological setting, since data often does not exist for impactful latent variables. For instance, when trying to learn the parameters for a simple SIR (Susceptible-Infected-Recovered) model, researchers likely would not have access to any statistics regarding the proportion of the population that is currently susceptible. Instead, they would need to infer this value based on data like the number of infections reported, leading to additional uncertainty and worse estimates. However, not all latent variables are created equal, which begs the question: what data is most important? In this project, we will explore how different unknown ODE system states influence the learnability of neural networks in an epidemiological context through the analysis of their performance estimating the ODE system parameters. By determining which states are hardest to learn in a SEIRM model, we can make informed decisions about how to allocate resources to increase our measurements over at-risk populations in order to better forecast epidemics.

2 RESPONSE TO MILESTONE COMMENT

The main response we got from our milestone was that we were behind schedule, and on our first response it was made clear that we needed to get into contact with Alex. We tried to get into contact with Alex but were unable to reach him. This was somewhat of a hindrance to our projects progress since we had a lot of trouble dealing with the EINN that was provided. It would have been very helpful to have been able to consult with someone who had worked with it but we were unable to get into contact with them so we needed to shift the plan for our project. To do this we switched to testing on a more generic Neural Network that was based on the design of the EINN. This stayed within our focus of trying to analize latent variables and the learnability of neural networks but could have benefited from more specialization within the domain of epidemiology.

3 RELATED WORK

3.1 ODEs and Epidemiology

Mathematical models have long been used to understand and predict the spread of diseases throughout a community. In Becker's 1978 work *The Uses of Epidemic Models* he goes over the history of some basic models that use factors like Susceptible population, Latent population, Infected population, and Recovered population, a similar structure to the basic SIR model [1]. He explains how these models can be used to both explain disease data and also can be used to provide insight into how diseases spread. They also allow a way to demonstrate how behavior between individuals might impact the spread of the community through the way people's behaviors can affect values like how much contact they have and how likely contact is to spread the disease between people. He also explores the insights different models have given us, such as how stochastic models have demonstrated that disease spread can vary due to chance fluctuations. They have also demonstrated the important concept of the epidemic threshold theorem where in large populations epidemics will either be small and die out or large epidemics which grow to huge proportions, and there will be a lack of epidemics with a medium size.

Furthermore, researchers have been using mathematical models to help understand the spread of ongoing infections for a long time. In his 1989 work Castillo-Chavez looks through the history of people using mathematical models to track and make predictions about the AIDS crisis [2]. In this work he discusses many of the problems researchers face as data for the AIDS crisis can be difficult to obtain. Models require parameter estimation and in real world situations this can be more challenging as data is not always reliably collected and some data can take long periods of time to come out. For example people who get AIDS may have the disease incubate for a long time so the time it takes to find out they are sick is not in the scale of a few days, like with COVID or the flu, but might instead be on the scale of months. There can also be issues with people choosing not to report so researchers do not know if they have accurate numbers. Despite these issues, mathematical models have been some of the best tools for predicting the short term spread of AIDS.

3.2 Epidemiological Modeling with Deep NNs

More recently, researchers have turned to deep learning for modelling pandemics; however, they have faced many challenges. There are multiple factors that contribute to the difficulty of learning the parameters of an ODE system in the context of modeling disease. Two situations where these factors can be observed are modeling in a data-scarce regime and a regime in which the data have noisy labels. One approach that can be utilized to address both of these difficulties is transfer learning of novel domains utilizing deep neural networks, which are trained in related domains for the problem of forecasting future incidences of a disease [3]. In this paper, the authors present the problem of predicting future weighted influenza-like-illness (wILI) counts over a set of regions. They have access to historical influenza-like-illness data and a deep neural network trained for forecasting on it, but they must adapt this previous work to account for the new effects of COVID-19. COVID-19 presents similar symptoms as influenza and therefore makes new data collected noisier. Furthermore, there is a relatively limited amount of well-labeled COVID-19 data when compared to historical wILI data.

First, the authors propose a network architecture called CALINET that learns to combine knowledge derived from wILI data with the new COVID-related signals. Then, the authors present a technique utilizing transfer learning that controls which knowledge is transferred to the new model according to the quality of the source model's predictions. An important strength of this paper is that it demonstrates that this knowledge distillation approach between neural networks is able to model trends unique to the new COVID-influenced ILI data. However, a weakness arises in the architecture that is employed for learning the dependencies between regional patterns of infection and the exogenous COVID-related signals: the GRU recurrent neural network. This architecture is particularly susceptible to disregarding long-term dependencies as the influence of the signal from initial sequence elements decays exponentially over the course of calculating an output. Deep neural networks that incorporate a self-attention layer in every module have recently been proposed and could be more effective here. In our project, we will similarly model disease propagation with a recurrent architecture, but we will incorporate the self-attention mechanism to overcome this limitation. Overall, this paper demonstrates the effectiveness of the latent neural network states as both training data and predictions in the context of modeling disease, which will be relevant to our project.

The authors of the EINN paper from which our project is based uses a new class of Physics Informed Neural Networks [6]. Physics informed neural networks are not a new concept and have been explored by teams like Yang et al. in their B-pinns: Bayesian physics informed neural networks for forward and inverse pde problems with noisy data[8]. In this paper they explored using Physics informed neural networks (PINN) to solve PDE problems. To improve upon these PINNs the team proposed adding a Bayesian aspect which allowed them to better quantify the uncertainty, and improve the accuracy.

A related problem for modeling epidemics with neural networks is the question of incorporating data from multiple modalities into the training process. Often, the richness of a dataset can be augmented by signals representing the same underlying content but from a different source medium. In the problem of forecasting, one can utilize data from multiple modalities, probabilistically encode them with deep neural networks, and combine their representations with a cross-attention layer before passing the result through a decoder to obtain a prediction of some state in at future timesteps [4]. This method, called "CAMul," has been shown to jointly model different datastreams, the dependencies between them, and dynamically select the most relevant samples, resulting in modeling of various time-series datasets at accuracy levels exceeding previous states-of-the-art. A key strength of this approach is the agnosticism of the architecture towards the specific modalities used in the modeling due to the latent encoding layer. One potential weakness of the paper is the explainability of the dynamic view selection module. While it is important to de-emphasize signals that are less relevant to the modeling task at hand, doing so with a cross-attention module may make this process opaque. It may be helpful to statistically analyze different input streams over time to determine which are less helpful and then globally downweight their significance. That being said, this paper is relevant to our project due to its philosophical similarity with EINNs, which incorporate both raw data inputs and mechanistic models into their training process.

Wang et al. have worked with synthetic data to train neural networks in an epidemiological context already. This team proposed DEFSI(Deep Learning Based Epidemic Forecasting with Synthetic Information)[7]. This proposed method had greater generalizability and physical consistency, and it outperformed in both high and low resolution. In short term state level Influenza-like illness prediction the proposed model matched or outperformed state of the art methods. In high resolution forecasting at the country level, the proposed method greatly outperformed existing methods. The success of this proposed model demonstrates the value that synthetic data can have when it comes to predictions in an epidemiological context. This paper provides significance to our project due to the validation it provides to using synthetic data which is what a core part of our project relies upon.

With the spread of COVID-19, an even greater emphasis has recently been placed on using data to forecast an emerging pandemic in real time. For instance, the DeepCOVID framework [5] proposed by Rodríguez et. al. uses deep learning to deal with large amounts of heterogeneous data in order to accurately estimate the proportion of the population that would become infected with COVID-19. In addition to this prediction module, DeepCOVID contains a data module, which helps standardize information coming from a wide range of sources, and an explainability module, which can be used to assist policymakers in understanding the current trends in the pandemic. However, while DeepCOVID and other work within this space have proven to be effective at forecasting COVID-19 infectivity rates within the short term, their accuracy often begins to dwindle when dealing with longer-term estimates.

3.3 Hybrid Approaches

A 2022 paper by Rodríguez and others tries to solve this problem through the use of Epidemiologically-Informed Neural Networks (EINNs) [6]. This neural network architecture leverages both the mathematical backing of mechanistic models, as well as the ability of AI models to ingest heterogeneous information, to accurately learn how a disease spreads. The basic premise of EINNs is that data generated using epidemic mechanistic models can be used to supervise the training of neural networks, allowing the networks to learn hidden dynamics of the epidemic while maintaining a general framework. In contrast to the methods discussed above, the ODE equations are not numerically solved during training, but rather directly estimated via supervision.

The main benefit of this approach is that the system does not need to be completely observable: if, for instance, there is no data regarding the population of susceptible individuals, it can simply be omitted. However, a major downside is that given complete information, estimating the parameters of the ODE equations directly would likely lead to better results than this approach; by indirectly estimating the system dynamics, we do not take full advantage of our knowledge of the underlying mathematical model. Nevertheless, this ability to omit unknown data is the key feature that will allow us to explore how the absence of certain variables influences learnability.

4 PROBLEM FORMULATION

In this project, we build a Neural Network inspired by the EINN work by Rodríguez and others [6] in order to study what data is most important when training neural networks for epidemiological models. More concretely, we want to determine the minimum sufficient population data needed to accurately predict the dynamics of various compartmental models, as well as which data is the most critical for generating a high-quality estimation. For example, if a pandemic's spread follows a SIR model, we want to know whether we will be able to accurately forecast the disease by relying only on "infected" and "recovered" population data. Answering this question will allow policymakers and other researchers to make more informed decisions regarding which data to focus on collecting, particularly in the early stages of any future pandemic.

The current state of the art, EINNs, are equipped to learn in settings where the networks of the time module are trained with information about each of the S, E, I, R, and M curves. However, in the wild, these values might be difficult to ascertain. Our intuition is that our experiments could lead to an improvement in the state of the art by learning a system under a greater degree of uncertainty (hidden states), making them more robust to real-life settings.

5 METHODOLOGY

5.1 The SEIRM Model

Because of the time constraints for this project, we will focus our efforts on analyzing the SEIRM compartmental model. The SEIRM model is a fairly common ODE model in epidemiology that has a Susceptible state, an Exposed state, an Infected state, a Recovered state, and a Mortality state. It also has four parameters β, α, γ, and μ, where β is the infectivity rate, $1/\alpha$ is the mean latent period for the disease, $1/\gamma$ is the mean infectious period, and μ is the mortality rate. The ordinary differential equations describing the behavior of this model are:

$$\frac{dS_t}{dt} = -\beta_t \frac{S_t I_t}{N} \qquad \frac{dE_t}{dt} = \beta_t \frac{S_t I_t}{N} - \alpha_t E_t$$

$$\frac{dI_t}{dt} = \alpha_t E_t - \gamma_t I_t - \mu_t I_t \qquad \frac{dR_t}{dt} = \gamma_t I_t \qquad \frac{dM_t}{dt} = \mu_t I_t$$

SEIRM is particularly useful for analyzing diseases with long incubation periods, since data about exposed individuals is much easier to collect before they become infected. As a result, it has seen a significant amount of use modeling the COVID-19 pandemic [6]. In addition to its popularity, we have chosen SEIRM because it is complex enough for us to see the effects of missing compartmental data, yet simple enough for us to understand quite intuitively. We hope that the results we see for SEIRM will extend to other common compartmental models.

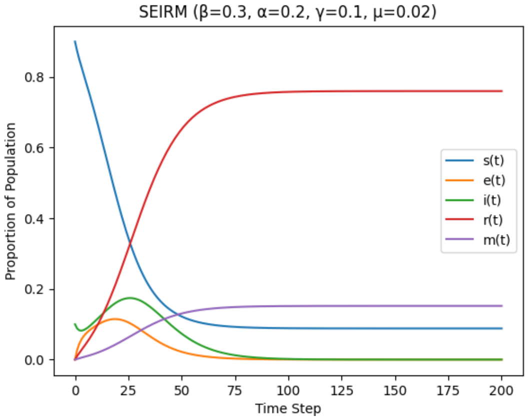

Figure 1: SEIRM Curves with Fixed Parameters

5.2 Data Collection

The first stage of our project was synthetic data generation. We used the SEIRM model to generate synthetic time-series data for multiple parameter combinations and record the S, E, I, R, and M values. In Figure 1, we show sample SEIRM curves over 200 time steps when $\beta_t = 0.3$, $\alpha_t = 0.2$, $\gamma_t = 0.1$, and $\mu_t = 0.02$ for all t. In Figure 2, we show how the SEIRM curves chance when β_t and γ_t are not constant, but instead vary with time.

Clearly, the behavior of these curves can become quite complex as we change $\beta_t, \alpha_t, \gamma_t$, and μ_t. Because of this, we have chosen to consider only a small subset of parameter time series types in our dataset collection. Firstly, we will allow parameter time series to take on constant values, such as the curves in Figure 1. Next, we will allow parameter time series to be generated from a linear function, such as $\gamma(t)$ in Figure 2. Lastly, we will allow parameter time series to be generated from a logistic function, such as $\beta(t)$ in Figure 2.

Given some initial parameter value x_0, final value x_f, and total time series length T, we can generate the time series $\{x_t\}_{t=0}^{T}$ for the parameter x using the following formulas. In the constant case, we have

$$x_t = c$$

for some constant c. In the linear case, we have

$$x_t = a + \frac{b-a}{T} t$$

In the logistic case, we have

$$x_t = a + \frac{b-a}{1 + \exp\left(c - \frac{2c}{T} t\right)}$$

for some constant c, where a larger c corresponds to a sharper change from the initial value to the final value. While these classes of generating functions do not cover all possible cases, we believe that the datasets we can create using their combinations will provide us with diverse enough data for our analysis of EINNs.

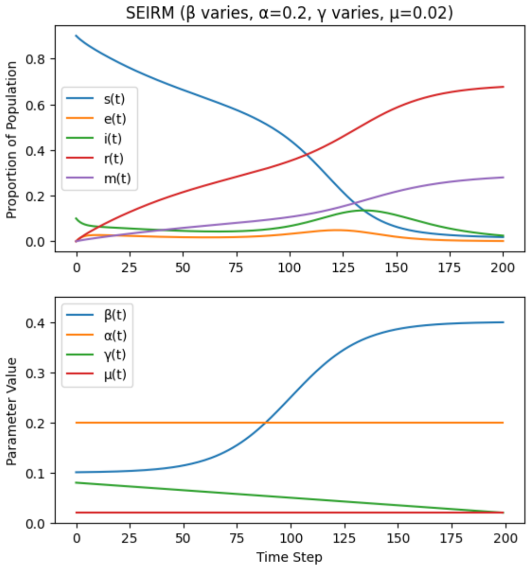

Figure 2: SEIRM Curves with Time-Varying Parameters

5.3 Model Predictions

The next phase is to use a Neural Network to generate predictions of the S curve based on the E, I, R, and M curves. We will then repeat this process with the E, I, R, and M compartments respectively. Once we have all of our predicted time series curves we will compare those to the ground truth curves generated by the ODE model and calculate their respective accuracies.

The Neural Network we will be using will be based off of the EINN by Rodríguez and others. What our Neural Network will be doing is taking in the values for a window of previous time steps and trying to predict the rate of change of the unknown compartment at the current time step. We use this prediction to generate a predicted curve for the hidden compartment. Once we have a predicted curve we compare that against the known values to calculate the accuracy.

Because both our synthetic and Neural Network generated curves consist of discrete values at the same time steps, we can use the sum of squared errors formula as our loss function:

$$L = \sum_{i} \left(y_i - f(\vec{x}_i)\right)^2$$

In addition to quantitatively analyzing the learned curves' accuracies, we qualitatively examine the resulting curves.

5.4 Model Explainability

After receiving feedback from our project proposal, we have expanded the scope of our project to examine not only learnability but also explainability of the neural network modules modeling our systems. One direction we are interested in is uncovering the relationship between input data and output activations, specifically which input features dominate the output signal of the time module and thus "explain" the decision-making process of the neural networks we train. Given that we are in a regression problem domain, the Integrated Gradients method for neural network explainability is a natural fit [7]. The general idea is, given a trained neural network F and an input $x \in R^n$, find the most relevant input features by summing the difference between the gradient of the input and the gradient of informationless baseline $x' \in R^n$ over a series of linear interpolations between x and x'. Explicitly, given the interpolation quantity α, the integrated gradient along the i^{th} dimension of x is given by:

$$IntegratedGrads_i(x) ::= (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} \, d\alpha$$

Note that this method depends on proper selection of the informationless baseline so that gradient differences are fully explained by the original input features. For our experiments, we will utilize the zero input vector.

We selected this approach because it exhibits several favorable properties in the context of our setting. First, as the method is invariant to implementation, it can be applied to any differentiable function, making it useful for the neural network architecture of the time module. Second, it is provably sensitive to all differences between the input and the baseline that result in different outputs. Both of these qualities are highly desirable for attribution.

After obtaining the results of this method, we will display which parts of the input where the most influential in determining the output of the model. This will be shown by a graph where the brighter blocks represent input features that had the greatest impact on the decision of the output.

6 EXPERIMENTS & RESULTS

6.1 Experimental Setup

We are hoping to answer the following two questions with our experiments. The first is: given a SEIRM ODE system, which of the epidemiological states over time is hardest to learn based on information from the others? The second is: given quantitative results from the previous question, can we qualitatively relate any parts of the input to the degree of success of the neural network that captures the latent dynamics of the system?

We begin by analyzing how the Neural Network performs when $\beta_t = 0.3$, $\alpha_t = 0.2$, $\gamma_t = 0.1$, and $\mu_t = 0.02$ for all t. First, we hide the S_t data and train on the E_t, I_t, R_t, and M_t data. We then compare the ground-truth synthetic S_t data with the model-generated data. We repeat this process, hiding each of E_t, I_t, R_t, and M_t while training on the remaining.

Next, we repeat the above procedure for 25 more randomly chosen sets of $\beta_t, \alpha_t, \gamma_t$, and μ_t. These datasets were generated based on the methods described in our prior data collection section. The random generation of these values provide us with good basis to be able to generalize.

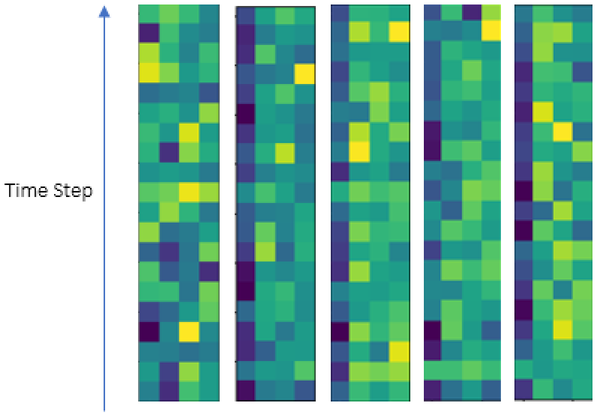

Figure 3: Integrated Gradients

6.2 Results

The table below displays the losses when each of the compartments is covered:

Curve Covered	Average Loss (Over 25 Datasets)
$S(t)$	0.0057
$E(t)$	0.0029
$I(t)$	0.0091
$R(t)$	0.0106
$M(t)$	0.0064

Here, a lower loss means that the associated curve is easier to accurately predict. From this table, we can tell that not all curves are equally as predictable given the data of the other curves. Furthermore, infected and recovered population data is significantly more important than the remaining compartmental data, while exposed population data seems to be the easiest to predict given the remaining four curves.

6.3 Model Explainability Results

In Figure 3, we see the result of applying the Integrated Gradients algorithm on our trained model:

This figure contains five sub-images, which correspond to the model's analysis when the $S(t)$, $E(t)$, $I(t)$, $R(t)$, and $M(t)$ curves, respectively, are covered. Each sub-image represents the importance of different model input features, where the horizontal axis coordinate corresponds to the input data curve and the vertical axis corresponds to the number of time steps in the past the input data is. A higher intensity pixel (i.e., more yellow) means that the feature is more important, while a lower intensity pixel (i.e., more purple) means that the feature is less important.

In the sub-image corresponding to $S(t)$ being covered, we see that the $R(t)$ curve is by far the most important in recovering $S(t)$. However, $R(t)$ is not nearly as critical in recovering $E(t)$, $I(t)$, or $R(t)$. On the other hand, $M(t)$ is important in recovering $E(t)$, $I(t)$, and $R(t)$. $S(t)$ seems to have very little effect in recovering any other curves, while $E(t)$ and $I(t)$ are helpful in recovering almost every curve other than themselves.

7 CONCLUSION & DISCUSSION

One important main take away from our work is that not all aspects of the SEIRM curve are equal in terms of ease to predict and importance to predicting other factors. We found that Infected and Recovered data were harder to predict than the other categories, and we found out that the Susceptible curve is not as relevant when it comes to trying to predict other curves. This is important for real world applications since it could inform policy makers in a real world scenario which aspects of data are the most important to collect when trying to forecast how an epidemic might grow.

It is unclear why the R curve proved to be the hardest to recover from information about the other curves. Our hypothesis is that, because R(t) tends to be of much higher magnitude than the other curves as t grows, the test loss was dominated by larger differences in the predicted R(t) and the ground truth R(t). For R(t), the network had to both fit the general direction of the curve and increase the magnitude of the guess relative to the inputs.

Our work provided interesting results in the effect of different parts of the SEIRM curve on Neural Network efficacy. Unfortunately the scope of our project was limited, but this does allow for potential room for future work to build off of. One possible direction future researchers could take this work is in analyzing how other variables like level of noise in the input data for the curves effects how accurately different grouping perform. Perhaps different parts of the curve are more or less resistant to noise, and if this is the case this might influence what real world data collection is deemed most important when there are varying levels of reliability. Another direction to go would be to try multiple combinations of variables to cover. For example see how well a neural network could predict S and I when both of them are hidden. This would also be useful for real world applications since often there may be multiple variables that the underlying data is not accessible. Since this research was done on synthetic data another interesting direction future researchers could go would be to see how these results would match up with tests done on real world data. It would be very useful to see if the theoretical results hold up to real world practice or if there are elements that are not being included which limit the usefulness of approaches like these. Another direction researchers could take this would be to analyze the Neural Network architecture and see how that influences the data. Researchers could plug in a variety of specialized EINNs or other Neural Networks from other domains to see if that has a deciding factor on which curves are most important.

8 REFERENCES

[1] N. Becker. The use of epidemic models, Biometrics, 35 (1978), pp. 295–305.

[2] C. Castillo-Chavez, ed. Mathematical and Statistical Approaches to AIDS Epidemiology,Lecture Notes in Biomath. 83, Springer-Verlag, Berlin, 1989.

[3] Alexander Rodrıguez et al. Steering a Historical Disease Forecasting Model Under a Pandemic: Case of Flu and COVID-19. In AAAI, 2021.

[4] Harshavardhan Kamarthi et al. CAMul: Calibrated and Accurate Multi-view Time-Series Forecasting. arXiv, 2021.

[5] Alexander Rodríguez et al. DeepCOVID: An Operational Deep Learning-driven Framework for Explainable Real-time COVID-19 Forecasting. In AAAI, 2021.
[6] Rodríguez, A., Cui, J., Ramakrishnan, N., Adhikari, B., and Prakash, B. A. (2022). EINNs: Epidemiologically-Informed Neural Networks. arXiv preprint arXiv:2202.10446.
[7] Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. Proceedings of the 34th International Conference on Machine Learning - Volume 70, pages 3319-3328
[8] Wang, L., Chen, J., and Marathe, M. (2019, July). DEFSI: Deep learning based epidemic forecasting with synthetic information. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, No. 01, pp. 9607-9612).
[9] Yang, L., Meng, X., Karniadakis, G. E. (2021). B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. Journal of Computational Physics, 425, 109913.