Couple of transmission models and deep learning techniques: brief introduction

Pengfei Song

Liam Lab in York University
Xi'an Jiaotong University

April 20, 2022

Supervised by Prof Jianhong Wu (York University)
Joint work with Prof Yanni Xiao (XJTU) and Shuangshuang Yin (XJTU)

1 Background

- Background
- Neural Networks Least to Know

2 Universal Differential Equations

- An Example of UDEs
- Story Behind

3 Applications

- Estimating time vary reproduction number
- Learning Unknown Human Behaviour Change Mechanisms
- Optimal Control

4 Why It Works?

Background

Epidemic models have proved to be a very powerful tool in guiding public health measures, learning from the past and preparing for the future. Nonetheless, modeling and controlling the emerging infectious disease such as COVID-19 remains a challenge due to the unknown mechanisms in transmission dynamics, for example,

- nonstandard incidence rate
- changing human mobility pattern
- shifting contact matrix
- evolution of virus
- ...

In this talk, I will introduce some answers from deep learning methods to handle the unknown mechanisms.

Neural Networks Least to Know

- input layer: $\mathcal{N}^0(\mathbf{U}) = \mathbf{U} \in \mathbb{R}^{d_{\mathrm{ta}}}$
- hidden layers: $\mathcal{N}^l(\mathbf{U}) = \sigma\left(W^l \mathcal{N}^{l-1}(\mathbf{U}) + b^l\right) \in \mathbb{R}^{N_l}$ for $1 \leq l \leq L-1$
- output layer: $\mathcal{N}^L(\mathbf{U}) = W^L \mathcal{N}^{L-1}(\mathbf{U}) + b^L \in \mathbb{R}^{d_{out}}$ where W^l is a Matrix or Tensor.

NN is composition of operators. Any abstract operator can be a layer, such as solution map of PDE. See more in NeurIPS 2020 tutorial: Deep Implicit Layers.

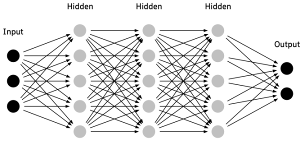


Figure: Scheme of deep neural network.

Neural Networks Least to Know: First Sentence

Neural Networks are universal apprimators. It can approximate unknown mappings and their derivatives. (Spectial "Taylor expansion")

Theorem (Allan Pinkus 1999 Acta Numer)

Let $m_i \in \mathbb{Z}^{d+}, i = 1, \cdots, s$, *and set* $m = \max_{i=1,\cdots,s} |m_i|$. *Assume* $\sigma \in C(\mathbb{R}^d)$ *and that* σ *is not a polynomial. Then a single hidden layer neural network:*

$$\mathcal{M}(\sigma) := \textbf{span}\left\{\sigma(\mathbf{w} \cdot \mathbf{U} + b) : \mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R}\right\}$$

is dense in

$$C^{\mathbf{m}^1,\ldots,\mathbf{m}^s}\left(\mathbb{R}^d\right) := \cap_{i=1}^s C^{\mathbf{m}^t}\left(\mathbb{R}^d\right)$$

.

Neural Networks Least to Know: First Sentence

Neural Networks are universal apprimators. It can approximate unknown nonlinear operators in Banach space, such as ellptic, nonlocal diffussion.

Theorem (Lu lu et.al. 2021 Nat Mach Intell)

Suppose that X is a Banach space, $K_1 \subset X$, $K_2 \subset \mathbb{R}^d$ are two compact sets in X and $\mathbb{R}^d$, respectively, V is a compact set in $C(K_1)$. Assume that $G : V \to C(K_2)$ is a nonlinear continuous operator. Then, for any $\epsilon > 0$, there exist positive integers m, p, continuous vector functions $g : \mathbb{R}^m \to \mathbb{R}^p, f : \mathbb{R}^d \to \mathbb{R}^p$, and $x_1, x_2, \cdots, x_m \in K_1$, such that

$$|G(u)(y) - < g(u(x_1), u(x_2), \cdots, u(x_m)), f(y) >| < \epsilon$$

holds for all $u \in V$ and $y \in K_2$, where $< \cdot, \cdot >$ denotes the dot product in $\mathbb{R}^p$. Furthermore, the functions g and f can be chosen as diverse classes of neural networks, which satisfy the classical universal approximation theorem of functions, for example, (stacked/unstacked) fully connected neural networks, residual neural networks and convolutional neural networks.

Neural Networks Least to Know: First Sentence

Neural Networks are universal apprimators.
So many universal approximators.

- Polinominal spaces $\{1, x, x^2, \cdots\}$
- Fourier expansion
- Chebyshev expansion
- Decision trees
- Non-parametric kernel methods like support vector machine, Gaussian process....

Why Neural Networks? UAs often face curse of dimensionality. NN can overcome curse of dimensionality and share bless of dimensionality. (Practice findings and with few theoretical results)

Neural Networks Least to Know: Second Sentence

Second: A blief in practice: for large neural networks, a local minima is enough and global minima often leads to over-fitting. It is a belief under theorectically exploring.

In a recent work (Kenji Kawaguchi 2016 arxiv), the authors theorectically proved some surprising results, Under Certain Conditions

- every local minimum is a global minimum
- every critical point that is not a global minimum is a saddle point
- there exist "bad" saddle points for the deeper networks (with more than three layers), whereas there is no bad saddle point for the shallow networks.

Neural Networks Least to Know: Second Sentence

Words to describe the learning ability.

- Deep neural networks can easily find good solutions or surrogates, and the surrogate doesn't need to be unique.
- DNN has unreasonable and counter-intuitive effectiveness, which helps to overcome curse of dimensionality and share bless of dimensionality.

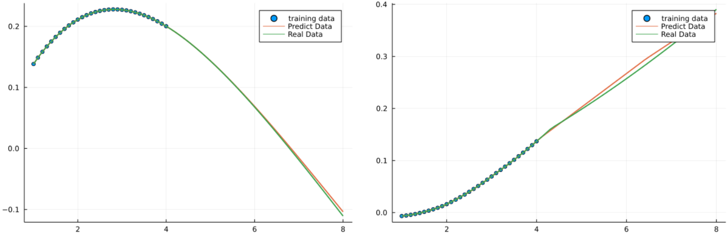


Figure: Left: the same neural network architechture with different parameters. Right: different neural network architechture.

Neural Networks Least to Know: Third Sentence

Feed on "big" data, Difficult to interpret and Weak generalization.

Trianing DNN by "big" data is an effective but unwise way. It costs a lot of resources like electiricity, which does harm to our earth.

- Alphago in 2016 costs about 35 million dollars
- Bert in 2018 has 0.3 billion paramters
- GPT-3 in 2020 has 175 billion parameters.
- Megatron-Turing in 2021 has 530 billion parameters.

It is AI but not human intelligence.

Neural Networks Least to Know: Third Sentence

Feed on "big" data, Difficult to interpret and Weak generalization.

- Newton's law, simple, it is science.
- Lorenz system, simple, it is science.
- SIR model, simple, it is science.
- Neural Networks, complex, it is black-box.

Researchers hate and love deep neural networks. The art of a good deep learning model is incorporating

Knowledge.

Join AI and Human Intelligence together. Universal differential equations is one way.

Outline

1 Background
- Background
- Neural Networks Least to Know

2 Universal Differential Equations
- An Example of UDEs
- Story Behind

3 Applications
- Estimating time vary reproduction number
- Learning Unknown Human Behaviour Change Mechanisms
- Optimal Control

4 Why It Works?

Universal Differential Equations

"Universal" means "universal approximators"
UDEs (proposed by Prof. Christopher Rackauckas, MIT, 2020(Christopher Rackauckas et.al. 2020 arxiv) are initial value problems with the following forms:

$$u' = f_{\theta_2}(u, t, NN_{\theta_1}(u, t)), \tag{1}$$

where f is a known mechanistic model and NN denotes the missing or unknown terms, θ_1 and θ_2 are parameters of known mechanisms and neural networks, respectively, which can be estimated simultaneously.

An Example Describing UDEs

To better understand UDEs, give the following simple epidemiological example to investigate the nonstandard incidence rate:

$$\begin{cases} \dfrac{\mathrm{dS}}{\mathrm{dt}} = -\mathrm{NN}(I)S, \\ \dfrac{\mathrm{dI}}{\mathrm{dt}} = \mathrm{NN}(I)S - \gamma I, \end{cases} \tag{2}$$

where S, I denote suspected and infected individuals, $\mathrm{NN}(I)$ denotes the unknown force of infection.

An Example Describing UDEs

Generating data from

$$\begin{cases} \dfrac{\mathrm{dS}}{\mathrm{dt}} = -\beta S \exp(-\alpha I) I^k, \\ \dfrac{\mathrm{dI}}{\mathrm{dt}} = \beta S \exp(-\alpha I) I^k - \gamma I, \end{cases} \tag{3}$$

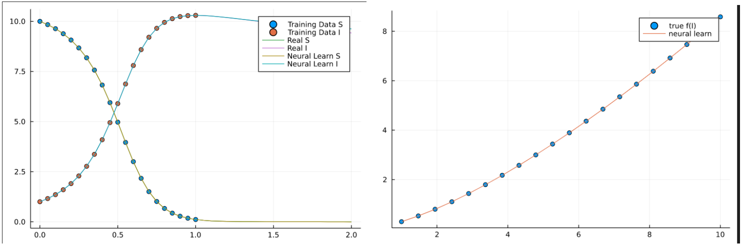


Figure: Using UDEs (2) to learn the nonstandard incidence rate in transmission model (3). Learn and generalize well

Universal Differential Equations

"Universal" means "universal approximators"

UDEs (proposed by Prof. Christopher Rackauckas, MIT, 2020(Christopher Rackauckas et.al. 2020 arxiv) are initial value problems with the following forms:

$$u' = f_{\theta_2}(u, t, NN_{\theta_1}(u, t)), \tag{4}$$

where f is a known mechanistic model and NN denotes the missing or unknown terms, θ_1 and θ_2 are parameters of known mechanisms and neural networks, respectively, which can be estimated simultaneously.

Story Behind: Neural Differential Equations

Neural differential equations 2018 (Chen Ricky Tianqi, 2018, NeurIPs), which were proposed before UDEs, inspired the ideas in UDEs and can be regarded as a special case of UDEs. Neural differential equations are initial value problems with the following form

$$u' = \mathrm{NN}_\theta(u, t), \tag{5}$$

where NN is a deep neural network receiving $[u, t]$ as input. Neural differential equations are still blackbox and no knowledge incorporated.

Story Behind: The Ideas behind Neural Differential Equations

Story behind neural differential equations.

- Inspired by powerfull ResNet

$$x(t+1) = x(t) + NN(x,t).$$

  Is continous-depth or "infinitely deep" ResNet possible?

- Redesigned sequential neural networks based on numerical shemes of differential equation.
- Revolutionary idea, training DNN as an optimal control problem, which extend the idea of backprogation (Book: Deep Learning) in differential programing to include adjoint sensitivity analysis (Cao et.al. 2003 SIAM J. Sci. Comput).

Story Behind: Universal Differential Equations

The story behind Universal Differential Equations.

$$u' = f_{\theta_2}(u, t, NN_{\theta_1}(u, t)), \tag{6}$$

- NDEs can approximate any sufficiently regular differential equation. However, it is defined without direct incorporation of known mechanisms. Needs "big" data, Difficult to interpret and Weak generalization.
- UDEs directly utilize mechanistic modeling simultaneously.
- UDEs are proved to be methods with good generalization and can be trained with less sample data (Christopher Rackauckas et.al. 2020 arxiv).

Story Behind: Universal Differential Equations

UDEs are beyond the extention of DNN or NDEs.

- The success behind UDEs are high-performance differential programing and scientific computation, espectially high-performance adjoint sensitivity analysis.
- UDEs can be more general and abstract

$$u' = Lu + f_{\theta_2}(u, t, \mathrm{UA}_{\theta_1}(u, t)) \tag{7}$$

  where L is some kind of abstract evolution operators such as ellptic, nonloncal diffussion in transmission models.
- *UA* can be other universal approximators such as decision trees, Chebyshev expansion, or Gaussian Process to handle uncertainty.
- Universal partial differential equations (UPDE), delayed differential equations (UDAE), Filippov systems, impulsive differential equations, stochastic differential equations(USDE).

Outline

1 Background
- Background
- Neural Networks Least to Know

2 Universal Differential Equations
- An Example of UDEs
- Story Behind

3 Applications
- Estimating time vary reproduction number
- Learning Unknown Human Behaviour Change Mechanisms
- Optimal Control

4 Why It Works?

Application One: Estimating time vary reproduction number

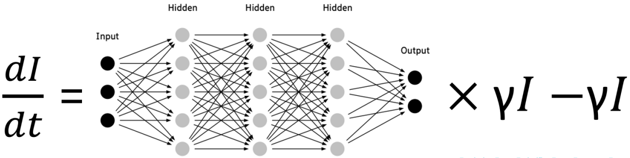
Represent effective reproduction number $\mathcal{R}_t$ as

$$\mathcal{R}_t = \mathrm{NeuralNetwork}_\theta(t, I), \tag{8}$$

and transmission model

$$\begin{cases} I' = \gamma \mathrm{NeuralNetwork}_\theta(t, I)I - \gamma I, \\ H' = \gamma \mathrm{NeuralNetwork}_\theta(t, I)I, \end{cases} \tag{9}$$

where $I(t)$ and $H(t)$ denote the number of infected individuals and accumulated confirmed cases at time t.



Application One: Estimating time vary reproduction number

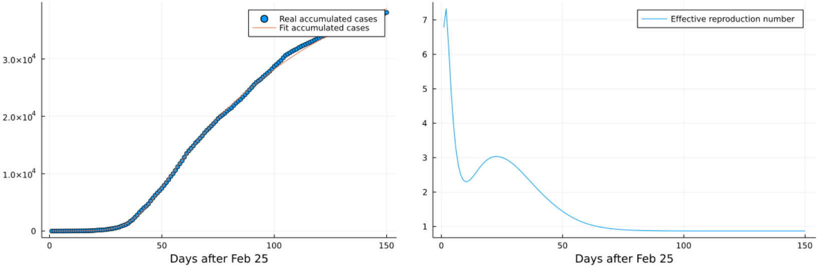


Figure: Left: Ontario's first wave COVID-19 case data. Right: effective reproduction number estimation by deep learning method.

Application One: Estimating time vary reproduction number

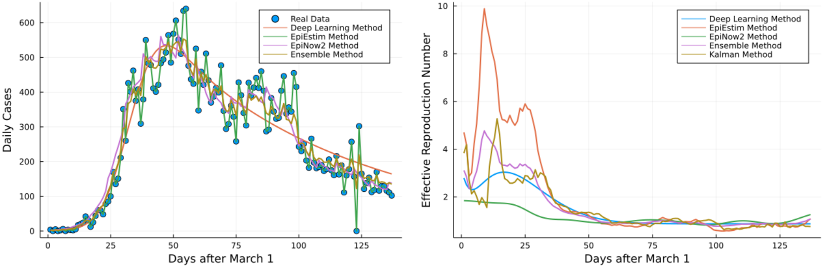


Figure: Left: Ontario's first wave COVID-19 case data fit by different methods. Right: effective reproduction number estimation by different methods.

Application One: Estimating time vary reproduction number

Methods	Data source	Smooth	Speed	Accuracy of $\mathcal{R}_t$
Deep Learning	Case data, infection period	strong	slow (3682s)	strong
State Space	Case data, infection period	weak	quick ($<$ 1s)	weak
EpiEstim	Case data, serial interval	weak	quick ($<$ 1s)	weak
EpiNow2	Case data, generation time, incubation period, delay distribution	normal	slow (2578s)	strong

Table: Comparison of different estimation methods: deep learning, state space, EpiEstim, EpiNow2 Methods. Smooth measures the data fitting abilities.

Application Two: Learning Unknown Human Behaviour Change Mechanisms

we will use the data of Ontario to fit the following neural differential equation model:

$$\frac{\mathrm{dS}}{\mathrm{dt}} = -\mathrm{abs}(NN(I,R))S/N,$$
$$\frac{\mathrm{dI}}{\mathrm{dt}} = -\mathrm{abs}(NN(I,R))S/N - \gamma I,$$
$$\frac{\mathrm{dR}}{\mathrm{dt}} = \gamma I,$$
$$\frac{\mathrm{dH}}{\mathrm{dt}} = abs(NN(I,R))S/N,$$

where $NN(I,R)$ denotes neural network to learn the human behaviour change, and H denotes accumulated cases.

Application Two: Learning Unknown Human Behaviour Change Mechanisms

To start with, learn the data by UDEs.

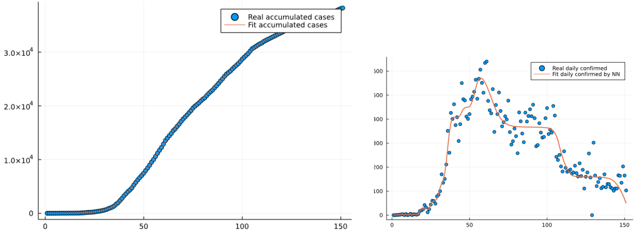
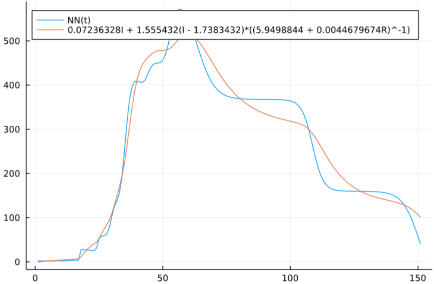


Figure: Learn Ontario first wave data by universal differential equations.

Application Two: Learning Unknown Human Behaviour Change Mechanisms

Use symbolic regression to find the simplest equation to fit $\mathrm{abs}(NN(I, R))$, and the equation found is kind of saturated function

$$\mathrm{abs}(NN(I, R)) \approx \frac{aI + b}{cR + d}.$$



Application Three: Optimal Control

Consider the following optimal control problem in *Bolza* form:

$$\begin{cases} \max\limits_{u(t)\in\Omega(t)} J = \int_0^T g(x,u,t)dt + \phi(x(T),T) \\ s.t. \quad \frac{\mathrm{d}x}{\mathrm{dt}} = f(x,u,t), x(0) = x_0, \end{cases} \tag{10}$$

where the functions $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^n$, $g : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$ and $\phi : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ are assumed to be continuously differentiable. By representing the optimal control function $u(t)$ as a neural network

$$u(t) = \mathrm{NeuralNetwork}_\theta(t,x) \tag{11}$$

receiving t and x as inputs.

Application Three: Optimal Control

$$\min_u \int_0^1 u^2 dt + I(1)^2, s.t. \quad I' = I - u, I(0) = 1.$$

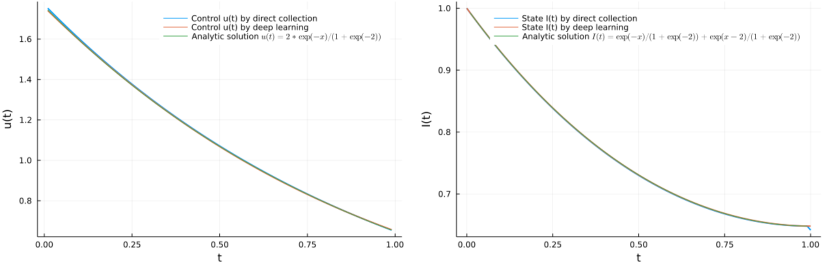


Figure: Left: control function. Right: state function.

Application Three: Optimal Control

$$
\begin{aligned}
\min \int_0^T & A * I(t) + u(t)^2 dt \\
s.t. \quad & S' = \Lambda - \beta S I - dS - u(t)S, \\
& E' = \beta S I - (d+\sigma)E, \\
& I' = \sigma E - (d+\gamma)I.
\end{aligned}
$$

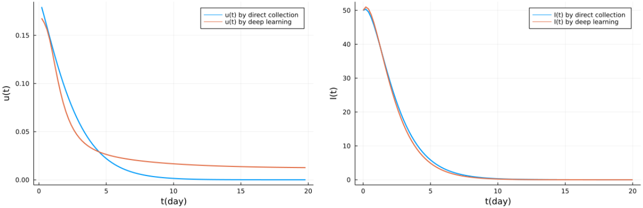


Figure: Left: control function. Right: state function. Comparison between direct allocation method and deep learning. Deep learning method find a good solution.

Application Three: Optimal Control

Methods	Direct method	Indirect method	HJB method	Deep learning
Transcriptions	Nonlinear programming problem (NLP)	Two point boundary value problem (TPBVP)	Dynamic programming	Parameter optimization
Trajectory or Parameter Optimization	Trajectory	Trajectory	Trajectory	Parameter
$u(x,t)$ or $x(u,t)$	-	$x(u,t)$	$u(x,t)$	$x(u,t)$
OtD or DtO	DtO	OtD	DtO	DtO and OtD
Using frequency	Most often	Often	Seldom	Seldom
Advantages	Mature Optimizers, easy to post, easy to solve	Accurate	Accurate	Flexible, extendable Bless of dimensionality
Disadvantages	Less accurate	Hard to post, hard to solve, initial guess	Curse of dimensionality	Theoretically under exploring,

Outline

1 Background
- Background
- Neural Networks Least to Know

2 Universal Differential Equations
- An Example of UDEs
- Story Behind

3 Applications
- Estimating time vary reproduction number
- Learning Unknown Human Behaviour Change Mechanisms
- Optimal Control

4 Why It Works?

Theoretical Engine: Optimization Theory in Infinite Dimensional Space

(Personal views)

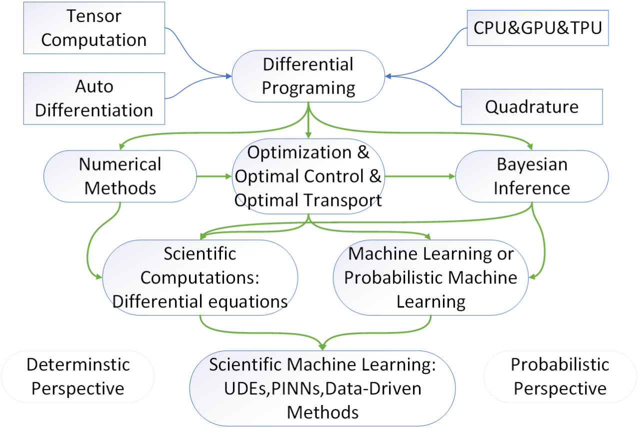
From the perspective of Optimization Theory in Infinite Dimensional Space (Variational Analysis, Optimization with PDE constraint, Optimal Control), it seems that couple of differential equation and deep learning is not a new story. Many mature theory and algorithms.

From the perspective of Deep Learning, Optimization Theory in Infinite Dimensional Space seems to be a new story.

We still have a long way to go to put this new engine on deep learning. More theory, more practice.

Practical Engine: Scientific Machine Learning Computation

The success behind machine learning is everything can and should be auto differentiable. (draw by myself, may be not that correct.)


Tensor Computation
Auto Differentiation
Differential Programing
CPU&GPU&TPU
Quadrature
Numerical Methods
Optimization & Optimal Control & Optimal Transport
Bayesian Inference
Scientific Computations: Differential equations
Machine Learning or Probabilistic Machine Learning
Determinstic Perspective
Scientific Machine Learning: UDEs,PINNs,Data-Driven Methods
Probabilistic Perspective

Publications on this topic

- Song, Pengfei and Xiao, Yanni and Wu, Jianhong. (2022) Methods coupling transmission models and deep learning. Preprint.
- Song, Pengfei and Xiao, Yanni and Wu, Jianhong. (2022) Discovering first-principle behavior change transmission models by deep learning methods. Preprint.
- Yin, Shuangshuang and Song, Pengfei and Wu, Jianhong. (2022) Optimal epidemic control by deep learning techniques. Preprint.
- Song, Pengfei and Xiao, Yanni. (2022) Estimating time-varying reproduction number by deep learning techniques (Dedicated to Prof Jibin Li on his 80th birthday). JAAC.

Thanks!