Couple of transmission models and deep learning techniques: brief introduction

Pengfei Song

Xi'an Jiaotong University

October 21, 2023

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

1 Background

2 Universal Differential Equations
- Application One: Estimating time vary reproduction number
- Application Two: Learning Unknown Mechanisms
- Application Three: Optimal Control
- Application Four: Misinformation Project

3 Story Behind UDE
- Neural Networks
- PINN

4 Discussion

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
- wastewater early warning
- impact of misinformation on disease spreading (NLP, UDE, Epi Model)
- ...

Background

Unknown formula of force of infection due to virus evolution, change human behaviour, impact of misinformation, ...

$$\begin{cases} \frac{\mathrm{dS}}{\mathrm{dt}} = -S\beta I, \\ \frac{\mathrm{dI}}{\mathrm{dt}} = S\beta I - \gamma I, \end{cases} \qquad \begin{cases} \frac{\mathrm{dS}}{\mathrm{dt}} = -S * (\text{Neural network}), \\ \frac{\mathrm{dI}}{\mathrm{dt}} = S * (\text{Neural network}) - \gamma I, \end{cases}$$

Can we embed data driven model in mechanism driven model to represent, learn and finally discover the unknown mechanisms?

Yes! Data and Mechanism driven Model

Data driven methods to learn unknown mechanism

- Mechanisms can be characterized by 'functions', 'operators', 'Distributions', 'Stochastic Processes'. For example, $f(x) = exp(x)$ describes exponential growth.
- How to learn mechanisms? Find the function or surrogates of the function from data. Approximation! $exp(x) \approx 1 + x + x^2/2 + \ldots$ Think about Taylor expansion, Fourier expansion.

  Functions: Neural Networks, Random feature model (Reservior Computing, ELM), GPs, Kernel Methods (e.g., SVM), Polynomials, Decision Trees

  Operators: DeepOnets, Neural Operator,

  Distributions: GAN, VAE, Auto-regressive models, Normalizing Flows, Diffusion Models, Energy Based Models, Consistency models

  Stochastic Processes: infinitely deep bayesian neural network as neural SDE

Background

- Applications of "AI For Science" on Mathematical Epidemiology
- AI4S is a direction combining fundamental researches (Mathematics, Physics, Chemistry, ...) with machine learning techniques.
- New revolutionary scientific research paradigm (Data and Mechanism driven Model)
- More on AI4S: 2022 International Congress of Mathematician 60 Minutes Talk:
  Weinan E: A Mathematical Perspective on Machine Learning
- Scientific Machine Learning (SciML); Scientific Artificial Intelligence (SciAI)

Outline

Universal Differential Equations

"Universal" means "universal approximators" (neural networks, GPs, SVM, random feature models, ...)

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

$$\frac{du}{dt} = f_{\theta_2}(u, t, NN_{\theta_1}(u, t)),$$

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.

Universal ODE

Generate data from

$$\begin{cases} \frac{\mathrm{dS}}{\mathrm{dt}} = -S\beta\exp(-\alpha I)I^k, \\ \frac{\mathrm{dI}}{\mathrm{dt}} = S\beta\exp(-\alpha I)I^k - \gamma I, \end{cases} \tag{1}$$

Learn by

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

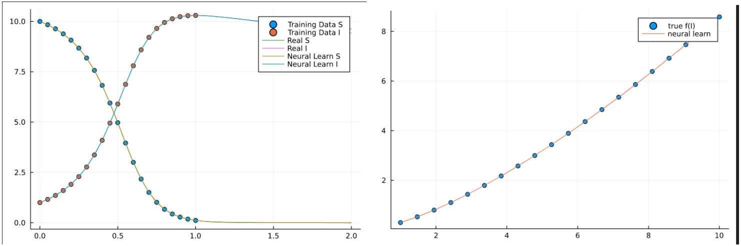
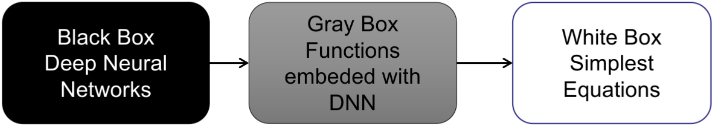


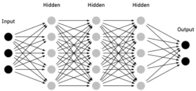
Figure: Using UDEs (2) to learn the nonstandard incidence rate in transmission model (1). Learn and generalize well

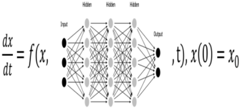
Black-box, Gray-box, White-box

How to recover the simplest function from deep neural networks?

Equation search methods.







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

[1] Song, Pengfei and Xiao, Yanni and Wu, Jianhong. (2023) Discovering first-principle behavior change transmission models by deep learning methods. One Chapter of Springer Book. Accepted

Equation Search Methods: Symbolic Regression

SR uses binary-tree to represent a function, and no particular formula is provided as a starting point. SR uses genetic programming, bayesian methods and deep learning methods to discover the simplest equations.

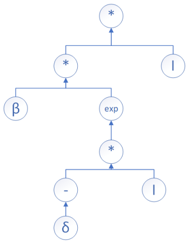


Figure: Expression tree of the function $f(I) = \beta \exp(-\delta I) I$

[1] Žegklitz J, Pošík P. Benchmarking state-of-the-art symbolic regression algorithms[J]. Genetic programming and evolvable machines, 2021, 22: 5-33.

Equation Search Methods: Sparse identification of nonlinear dynamic systems

SINDy applies a set of candidate functions $\mathbf{\Theta}(\mathbf{U})$ that would characterize the right-hand side of the governing equations, $u' = f(u) \approx \Theta(u)\Xi$, and estimate Ξ by sparse regression.

$$\begin{cases} \frac{\mathrm{dS}}{\mathrm{dt}} = -\beta SI, \\ \frac{\mathrm{dI}}{\mathrm{dt}} = \beta SI - \gamma I. \end{cases} \tag{3}$$

We choose the basic functions as

$$\mathbf{\Theta}([S, I]) = \left[S, I, SI, S^2 I, SI^2, S^2 I^2\right]$$

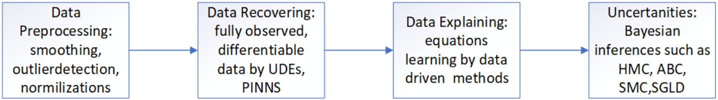
and now we use SINDy to discover the true equations.

[1] Brunton S L, Proctor J L, Kutz J N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems[J]. Proceedings of the national academy of sciences, 2016, 113(15): 3932-3937.

Two-step Learning-Explaining Methods

UDE: representing and learning the unknown mechanisms by neural networks

Equation Search: recover the simplest function from neural networks



Remark: recover the simplest function from neural networks is favored by biologists and mathematician, but NOT in line with the philosophy of deep learning. Improve the generalization ability is.

[1] Song, Pengfei and Xiao, Yanni and Wu, Jianhong. (2023) Discovering first-principle behavior change transmission models by deep learning methods. One Chapter of Springer Book. Accepted

Application One: Estimating time vary reproduction number

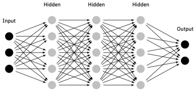
Represent effective reproduction number $\mathcal{R}_t$ as

$$\mathcal{R}_t = \mathrm{NeuralNetwork}_\theta(t, I), \tag{4}$$

and transmission model

$$\begin{cases} I' = \gamma \mathrm{NeuralNetwork}_\theta(t, I) I - \gamma I, \\ H' = \gamma \mathrm{NeuralNetwork}_\theta(t, I) I, \end{cases} \tag{5}$$

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

$\frac{dI}{dt} =$  $\times \gamma I - \gamma I$

[1] 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

Application One: Estimating time vary reproduction number

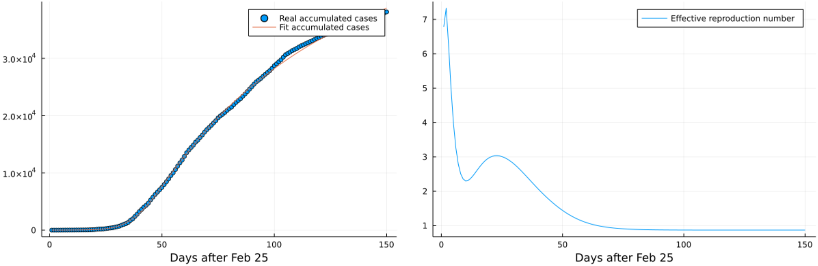
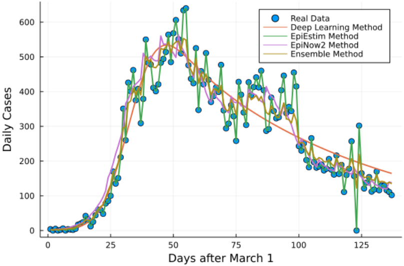


Figure: Left: Ontario's first wave COVID-19 case data. Right: effective reproduction number estimation by deep learning method.

Application One: Estimating time vary reproduction number



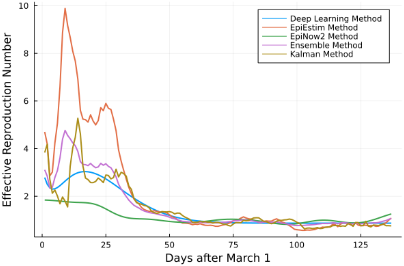


Figure: Left: Ontario's first wave COVID-19 case data fit by different methods. Right: effective reproduction number estimation by different methods.

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.

[1] Song, Pengfei and Xiao, Yanni and Wu, Jianhong. (2023) Discovering first-principle behavior change transmission models by deep learning methods. One Chapter of Springer Book. Accepted

Application Two: Learning Unknown Human Behaviour Change Mechanisms

To start with, learn the data by UDEs.

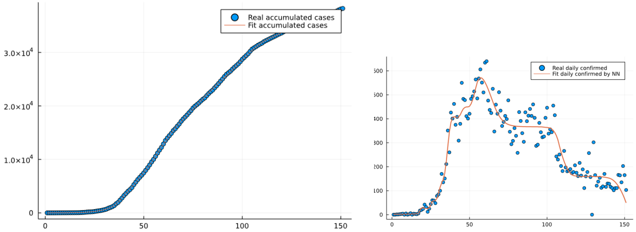


Figure: Learn Ontario first wave data by universal differential equations.

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

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

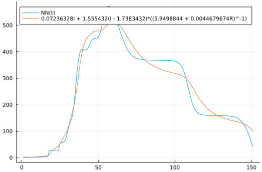


Figure: Symbolic regression to find the simplest equation to fit $\mathrm{abs}(NN(I, R))$.

[1] Song, Pengfei and Xiao, Yanni and Wu, Jianhong. (2023) Discovering first-principle behavior change transmission models by deep learning methods. One Chapter of Springer Book. Accepted

Application Three: Optimal Control

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

$$\begin{cases} \max\limits_{u(t)\in\Omega(t)} J = \int_0^T g(x,u,t)dt + \phi(x(T),T) \\ s.t. \quad \frac{\mathrm{d}x}{\mathrm{dt}} = f(x,u,t), x(0) = x_0, \end{cases} \tag{6}$$

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

$$u(t) = \mathrm{NeuralNetwork}_\theta(t,x) \tag{7}$$

receiving t and x as inputs.

[1] Yin, Shuangshuang and Song, Pengfei and Wu, Jianhong. (2023) Optimal epidemic control by deep learning techniques. Journal of Mathematical Biology.

Application Three: Optimal Control

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

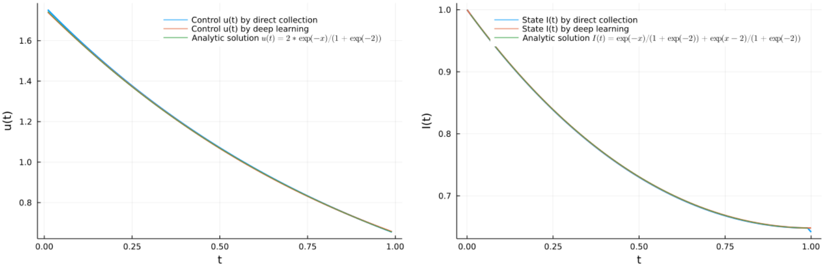


Figure: Left: control function. Right: state function.

Application Three: Optimal Control

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

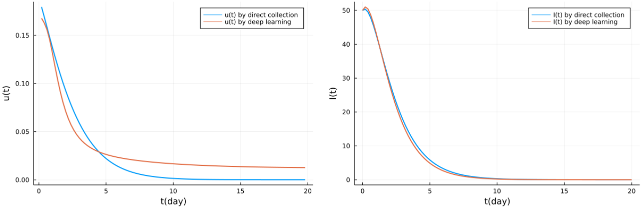


Figure: Left: control function. Right: state function. Comparison between direct allocation method and deep learning. Deep learning method find a good solution.

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,

Application Four: Misinformation Project

Question: the impact of misinformation on disease spread and vaccination decision?

- Information Data: classification of information(区分正确和错误信息): NLP(自然语言处理) such as Bert, ChatGPT.
- Evolution of correct and misinformation(未知的信息演化规律): Neural ODE
- Impact of misinformation on disease spread and vaccination decision(信息对传染病传播和疫苗接种未知的影响): UDE

Application Four: Misinformation Project

Disease Model

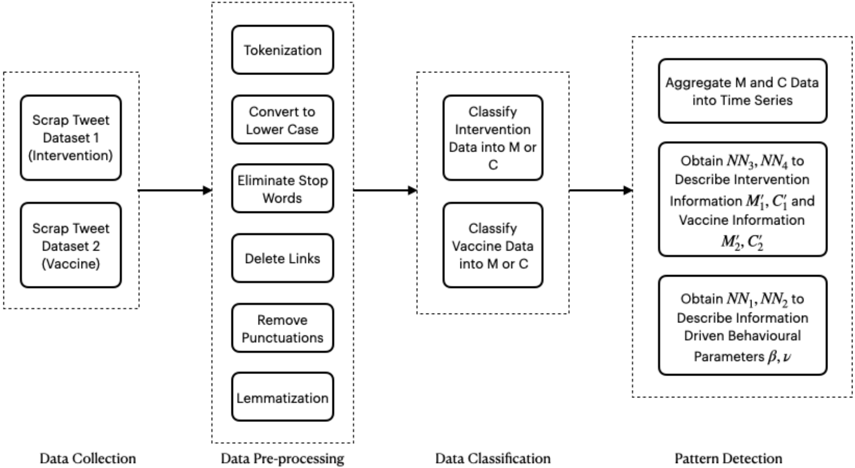
$$\begin{cases} S' &= -\beta_1 S\frac{I}{N} - \nu(t)S \\ V' &= \nu(t)S - \beta_2 V\frac{I}{N} \\ E' &= \beta_1 S\frac{I}{N} + \beta_2 V\frac{I}{N} - \sigma E \\ I' &= \sigma E - \gamma I \\ R' &= \gamma I \end{cases} \quad \text{with} \quad \begin{cases} \beta_1(t) &= NN_1(t, M_1, C_1, M_2, C_2 \\ \beta_2(t) &= (1-\epsilon)\beta_1 \\ \nu(t) &= NN_2(t, M_1, C_1, M_2, C_2 \end{cases}$$

Information Model

$$\begin{cases} M_1' &= NN_3(t, M_1, C_1)[1] \\ C_1' &= NN_3(t, M_1, C_1)[2] \end{cases} \qquad \begin{cases} M_2' &= NN_4(t, M_2, C_2)[1] \\ C_2' &= NN_4(t, M_2, C_2)[2] \end{cases}$$

We will investigate the relationship between the two sets of neural networks, NN_1 and NN_3 (for intervention), as well as NN_2 and NN_4 (for vaccine).

Application Four: Misinformation



Outline

Neural Networks

- 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 $1 \le l \le L-1$
- 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, solution of implicit function. See more in NeurIPS 2020 tutorial: Deep Implicit Layers.

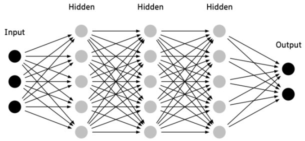


Figure: Scheme of deep neural network.

Neural Networks: Universal Apprimators

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:

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

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: Universal Apprimators

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: Universal Apprimators

Generative models (GAN, VAE, Auto-regressive models, Normalizing Flows, Diffusion Models, Energy Based Models, Consistency models) are universal approximators for distributions.

Continuous normalizing flow:

$$x' = NN(x, t), x(0) \backsim p$$

Why Neural Networks? Solving Curse of dimensionality

Universal approximators like Polinominal spaces, Fourier expansion , Chebyshev expansion, Decision trees , Gaussian process face curse of dimensionality (COD). Many techniques are needed to handle COD, such as sparsity, parallel computing.

However, NN is believed to some extent share bless of dimensionality (many practical findings and few theoretical results).

[1]Weinan E: A Mathematical Perspective on Machine Learning. 2022 International Congress of Mathematician 60 Minutes Talk

Why Neural Networks? Powerful Generalization, Implicit Regularization

- Escaping saddle point
- Sharpness aware
- Local minimum is enough: easily find good solutions or surrogates, and the surrogate doesn't need to be unique

[1] Du S S, Jin C, Lee J D, et al. Gradient descent can take exponential time to escape saddle points. Advances in neural information processing systems, 2017, 30.

[2] Wu L, Su W J. The Implicit Regularization of Dynamical Stability in Stochastic Gradient Descent. ICML, 2023.

[3] Foret P, Kleiner A, Mobahi H, et al. Sharpness-aware minimization for efficiently improving generalization. COLT, 2020.

Why Not Neural Networks Only?

Feed on "big" data, Difficult to interpret, . Trianing DNN by "big" data is an effective but unwise way.

- Alphago in 2016 costs about 35 million dollars
- GPT-3 in 2020 has 175 billion parameters.
- Megatron-Turing in 2021 has 530 billion parameters.
- Recent Persia can have 100 Trillion parameters.
- GPT3.5: 200 billion; GPT4 not known.

It is AI but not human intelligence!?

Why Not Neural Networks Only? Incorporating Knowledge

Feed on "big" data, Difficult to interpret.

- 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.

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:

$$u' = f_{\theta_2}(u, t, NN_{\theta_1}(u, t)), \tag{8}$$

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.

How to Train Universal Differential Equations?

The essence of training UDE is to solve the following abstract evolution equations constrained optimization problem or optimal control problem (or inverse problems or bayesian inversion problems):

$$\begin{cases} \min_{\theta} J = \mathbb{E} \int_0^T g(X, \mathrm{NeuralNetwork}_\theta(t, X), t) dt + \phi(X(T), T) \\ s.t. \quad F(t, X(t), X(\alpha(t)), \mathrm{UA}(t, X(t), X(\beta(t))), W(t)) = 0, t \in [0, T], \end{cases} \tag{9}$$

F denotes the abstract evolution equations such as stochastic partial functional differential equations.

How to Train Universal Differential Equations?

For ODE case, its essence is to solve the following optimal control problem:

$$\begin{cases} \min\limits_{\theta} J = \int_0^T g(x, \mathrm{NeuralNetwork}_\theta(t, x), t) dt + \phi(x(T), T) \\ s.t. \quad \frac{\mathrm{d}x}{\mathrm{dt}} = f(x, \mathrm{NeuralNetwork}_\theta(t, x), t), x(0) = x_0, t \in [0, T]. \end{cases} \tag{10}$$

How to Train Universal Differential Equations?

" backpropagation" for differential equations: Adjoint sensitivity analysis in optimal control theory (Cao et.al. 2003 SIAM J. Sci. Comput).

Theorem

Let

$$J(\theta) = \int_0^T e(y,t)dt, y' = m(y,\theta,t), y(0) = y_0,$$

where $\theta \in \mathbb{R}^k$ *and the functions* $m : \mathbb{R}^n \times \mathbb{R}^k \times \mathbb{R} \to \mathbb{R}^n$, $e : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$ *are continuously differentiable. Then we have*

$$\begin{cases} \frac{\mathrm{d}J}{\mathrm{d}\theta} = \int_0^T \lambda(t) m_\theta dt, \\ \lambda'(t) = -e_y - m_y \lambda(t), \lambda(T) = 0, \\ y' = m(y,\theta,t), y(0) = y_0. \end{cases} \tag{11}$$

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

$$u' = Lu + f_{\theta_2}(u, t, \mathrm{UA}_{\theta_1}(u, t)) \tag{12}$$

  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).

Universal DDE

Generate Data from the following SIR model with lag effect of media impact:

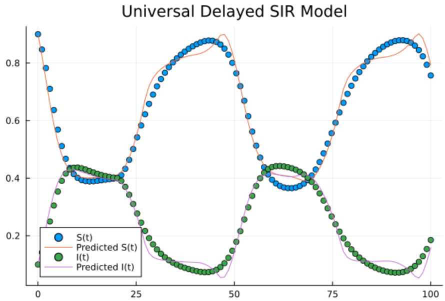
$$\begin{cases} \dfrac{\mathrm{dS}}{\mathrm{dt}} = \Lambda - S\beta I \exp(-\alpha I(t-\tau)) - dS, \\ \dfrac{\mathrm{dI}}{\mathrm{dt}} = S\beta I \exp(-\alpha I(t-\tau)) - dI - \gamma I, \end{cases} \tag{13}$$

where S, I denote suspected and infected individuals.

Universal DDE

Using Generated data to learn

$$\begin{cases} \frac{\mathrm{dS}}{\mathrm{dt}} = \Lambda - S\mathrm{NN}(I(t), I(t-\tau)) - dS, \\ \frac{\mathrm{dI}}{\mathrm{dt}} = S\mathrm{NN}(I(t), I(t-\tau)) - dI - \gamma I, \end{cases} \tag{14}$$



Universal Filippov System

Generate Data from the following switched SIR model:

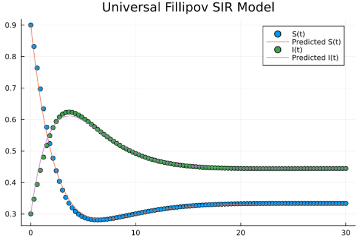
$$\begin{cases} \frac{\mathrm{dS}}{\mathrm{dt}} = \Lambda - \epsilon\beta SI - dS, \\ \frac{\mathrm{dI}}{\mathrm{dt}} = \epsilon\beta SI - dI - \gamma I, \\ \epsilon = 1.0(I \leq 0.3), \epsilon = 0.5(I > 0.3), \end{cases} \tag{15}$$

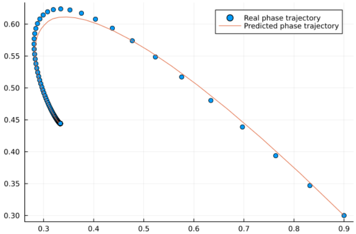
where S, I denote suspected and infected individuals.

Universal Fillipov System

Using Generated data to learn

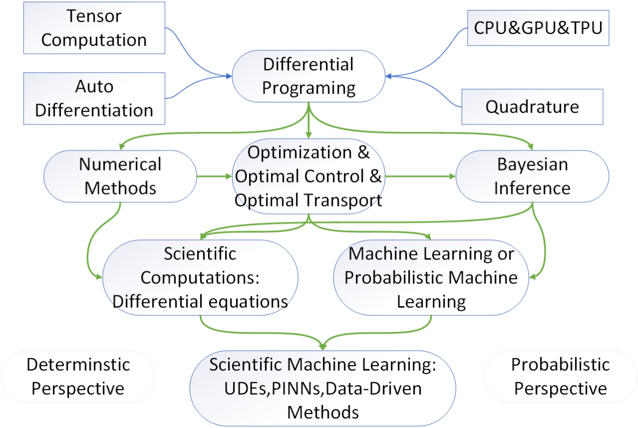
$$\begin{cases} \frac{\mathrm{dS}}{\mathrm{dt}} = \Lambda - \epsilon \mathrm{NN}(S(t), I(t)) - dS, \\ \frac{\mathrm{dI}}{\mathrm{dt}} = \epsilon \mathrm{NN}(S(t), I(t)) - dI - \gamma I, \\ \epsilon = 1.0(I \leq 0.3), \epsilon = 0.5(I > 0.3), \end{cases} \tag{16}$$





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.)



Practical Skills

How to avoid local minimum? Multiple Shooting

How to find a good initial value quickly? Collection Methods

Difference between PINN and UDE

- PINN: $x' = L_{\theta_1} x + f_{\theta_2}(x)$, $x = NN_{\theta_2}(t)$. Using DNN to SOLVE forward and inverse problems of differential equations.
- UDE: $x' = L_{\theta_1} x + f_{\theta_2}(x, NN_{\theta_2}(x, t))$.

Outline

Publications on this topic

- Song, Pengfei and Xiao, Yanni and Wu, Jianhong. (2023) Methods coupling transmission models and deep learning. Preprint.
- Song, Pengfei and Xiao, Yanni and Wu, Jianhong. (2023) Discovering first-principle behavior change transmission models by deep learning methods. One Chapter of Springer Book. Accepted
- Yin, Shuangshuang and Song, Pengfei and Wu, Jianhong. (2023) Optimal epidemic control by deep learning techniques. JMB.
- 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!