DEEP LEARNING OF PARAMETERIZED EQUATIONS WITH APPLICATIONS TO UNCERTAINTY QUANTIFICATION

Tong Qin, [1] Zhen Chen, [2] John D. Jakeman, [3] & Dongbin Xiu [2]

[1]University of Michigan [2]The Ohio State University [3]Sandia National Laboratories

Abstract

We present a data-driven method to learn unknown parameterized dynamical system from measurement of states of the system. And we apply it in uncertainty quantification (UQ).

- Method based on deep neural network (DNN).
- Do system prediction over long time for arbitrary parameter values.
- UQ analysis by evaluating solution statistics over parameter space.

Problem Setup

Consider an unknown parameterized system

$$\frac{d}{dt}\mathbf{x}(t;\boldsymbol{\alpha}) = \mathbf{f}(\mathbf{x},\boldsymbol{\alpha})$$

where $\mathbf{x} \in \mathbb{R}^d$ is state variable and $\boldsymbol{\alpha} \in \mathbb{R}^l$ are system parameters. We are interested in the solution behavior with respect to varying parameters. In UQ setting, parameters are random variables and equipped with a probability measure.

Assumptions

- Form of equation $\mathbf{f}(\mathbf{x},\boldsymbol{\alpha}) : \mathbb{R}^d \times \mathbb{R}^l \to \mathbb{R}^d$ is unknown.
- Trajectories of data is available, $\{\mathbf{x}(t^{(i)};\boldsymbol{\alpha}^{(i)},\mathbf{x}_0^{(i)}),\mathbf{x}_0^i\}$.

Goals

- We want to approximate the flow map using DNN.

$$\mathbf{x}(t;\boldsymbol{\alpha},\mathbf{x}_0) = \boldsymbol{\Phi}_t(\mathbf{x}_0,\boldsymbol{\alpha})$$

  which can be written as

$$\mathbf{x}(t;\boldsymbol{\alpha},\mathbf{x}_0) = \mathbf{x}_0 + \int_0^t \mathbf{f}(\boldsymbol{\Phi}_s(\mathbf{x}_0;\boldsymbol{\alpha}))ds$$

- In UQ setting, we want to use trained model to do uncertainty quantification analysis.

Method

First we present our neural network structure, then we talk about application for uncertainty quantification.

Neural Network Model Construction We define network model as

$$\tilde{\mathbf{x}}(t;\boldsymbol{\alpha},\mathbf{x}_0) := \mathbf{x}_0 + \mathbf{N}(\mathbf{x}_0,\boldsymbol{\alpha},t;\Theta)$$

where $\mathbf{N}(;\Theta)$ is a fully-connected neural network, Θ is its parameter set.

- This can be considered as a exact time integrator in term of flow map.

$$\mathbf{x}(t;\boldsymbol{\alpha},\mathbf{x}_0) = \mathbf{x}_0 + \int_0^t \mathbf{f}(\boldsymbol{\Phi}_s(\mathbf{x}_0;\boldsymbol{\alpha}))ds$$

- Define loss function

$$\mathcal{L}(\Theta) := \frac{1}{N_{data}}\sum_{i=1}^{N_{data}}\left\|\mathbf{x}_0^{(i)} + \mathbf{N}(\mathbf{x}_0^{(i)},\boldsymbol{\alpha}^{(i)},t^{(i)};\Theta) - \mathbf{x}^{(i)}(t^{(i)};\boldsymbol{\alpha}^{(i)},\mathbf{x}_0^{(i)})\right\|_2^2$$

Prediction and Uncertainty Quantification

Once the model is constructed, we can do prediction by using the model iteratively and do uncertainty quantification analysis over parameter space.

Prediction

- Given an initial condition $\mathbf{x}_0$, a time sequence $\{t_k\}$, and a system parameter $\boldsymbol{\alpha}$.
- Define $\tilde{\mathbf{x}}(t_0;\boldsymbol{\alpha}) = \mathbf{x}_0$, for $k = 0, \ldots$

$$\tilde{\mathbf{x}}(t_{k+1};\boldsymbol{\alpha}) = \tilde{\mathbf{x}}(t_k;\boldsymbol{\alpha}) + \mathbf{N}(\tilde{\mathbf{x}}(t_k;\boldsymbol{\alpha}),\boldsymbol{\alpha},t_{k+1}-t_k;\Theta^*)$$

Uncertainty Quantification

- Consider random variable $\boldsymbol{\alpha} \in I_{\boldsymbol{\alpha}}$, with density function $\rho_{\boldsymbol{\alpha}}$.
- Obtain statistical information of surrogate model. For example,

$$\mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{x}(t;\boldsymbol{\alpha})] \approx \mathbb{E}_{\boldsymbol{\alpha}}[\tilde{\mathbf{x}}(t;\boldsymbol{\alpha})] = \int_{I_{\boldsymbol{\alpha}}} \tilde{\mathbf{x}}(t;\mathbf{y})\rho_{\boldsymbol{\alpha}}(\mathbf{y})d\mathbf{y}$$

$$\mathrm{Var}_{\boldsymbol{\alpha}}[\mathbf{x}(t;\boldsymbol{\alpha})] \approx \mathrm{Var}_{\boldsymbol{\alpha}}[\tilde{\mathbf{x}}(t;\boldsymbol{\alpha})] = \int_{I_\alpha} [\tilde{\mathbf{x}}(t;\mathbf{y}) - \mathbb{E}_{\boldsymbol{\alpha}}[\tilde{\mathbf{x}}(t;\mathbf{y})]]^2 \rho_{\boldsymbol{\alpha}}(\mathbf{y})d\mathbf{y}$$

Illustration Example: Cell Signaling Cascade

Consider the mathematical model for autocrine cell-signaling loop. e_{1p}, e_{2p}, and e_{3p} denote the concentrations of the enzymes.

$$\frac{de_{1p}}{dt} = \frac{I}{1+Ge_{3p}}\frac{V_{\max,1}(1-e_{1p})}{K_{m,1}+(1-e_{1p})} - \frac{V_{\max,2}e_{1p}}{K_{m,2}+e_{1p}}$$

$$\frac{de_{2p}}{dt} = \frac{V_{\max,3}e_{1p}(1-e_{2p})}{K_{m,3}+(1-e_{2p})} - \frac{V_{\max,4}e_{2p}}{K_{m,4}+e_{2p}}$$

$$\frac{de_{3p}}{dt} = \frac{V_{\max,5}e_{2p}(1-e_{3p})}{K_{m,5}+(1-e_{3p})} - \frac{V_{\max,6}e_{3p}}{K_{m,6}+e_{3p}}$$

- This model contains 13 (random) parameters. The parameter space is high-dimensional and this is challenging for most existing methods.
- A parameter of particular interest is I, the tuning parameter. It affects the steady state solution.

Sampling Data and Training

- Sample initial conditions/state variables uniformly from $[0,1]^3$.
- Sample parameters independently and uniformly from a hypercube of ±10% around $K_{m,1-6} = 0.2$, $V_{\max,1} = 0.5$, $V_{\max,2} = 0.15$, $V_{\max,3} = 0.15$, $V_{\max,5} = 0.25$, $V_{\max,6} = 0.05$, $V_{\max,4} = 0.15$, and $G = 2$.
- For each initial condition and parameter, obtain 1 time step trajectory data.
- Train using FNN with 3 layers and 200 nodes each layer. Data set of size 300, 000.

Prediction and UQ Analysis

- Given initial condition x0 = (0.22685145, 0.98369158, 0.87752945) and time march forward 1, 400 steps.
- Calculate the mean and variance of the state variables with respect to the random parameters $K_{m,1}, K_{m,4}, V_{\max,2}$, and $V_{\max,5}$.
- Plot response curve of e_{3p} steady state with respect to the tuning parameter I.

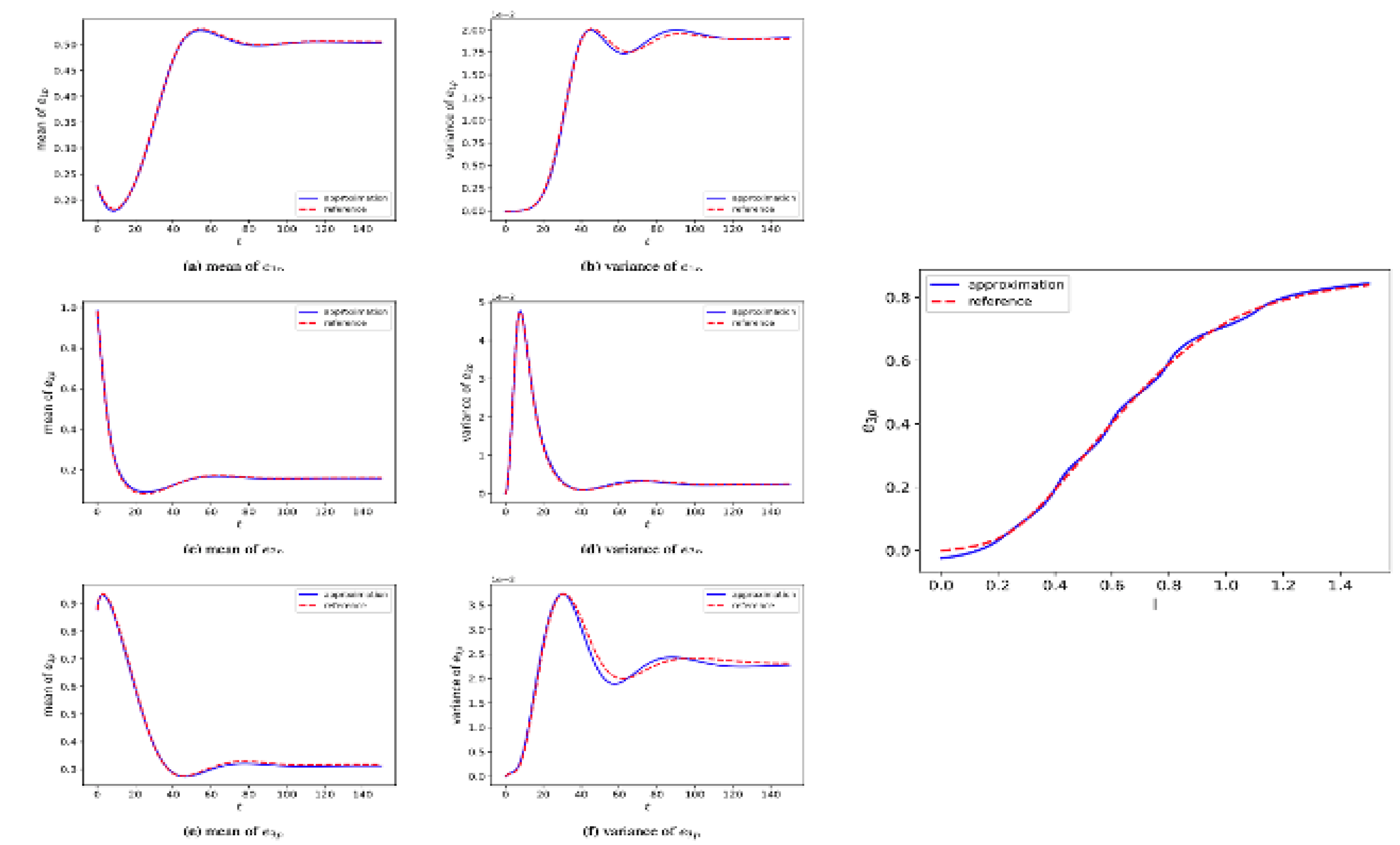


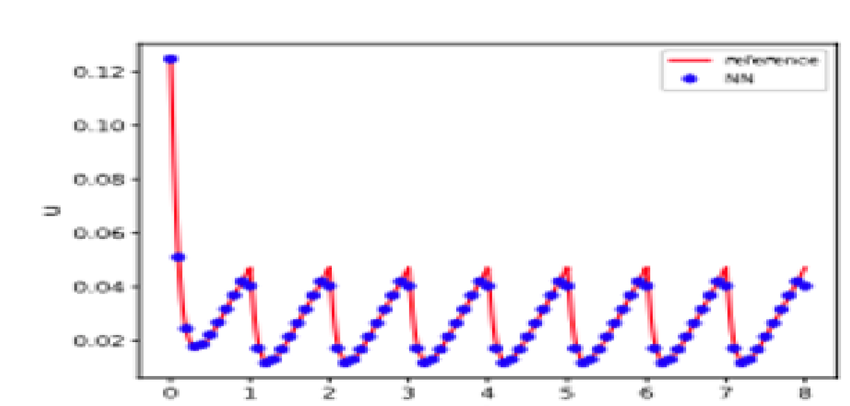
Figure 1:Mean and variance of concentrations (left) and steady state response curve of e_{3p} to I (right).

Extension to Non-Autonomous Dynamical Systems

We extend our method [1] which learn autonomous systems with (uncertain) time-invariant parameters to learning non-autonoumous systems with time-dependent parameters [2].

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{f}(\mathbf{x},\boldsymbol{\alpha}(t))$$

- Solution states of non-autonomous system depend on the entire history of the system, this renders most data-driven method inapplicable.



(a) Solution evolution at $x = 0.5$

$$u_t = u_{xx} + q(t,x), \quad x \in [0,1]$$
$$u(0,x) = u_0(x)$$
$$u(t,0) = u(t,1) = 0$$
$$q(t,x) = \alpha(t)e^{-\frac{(x-\mu)^2}{\sigma^2}}$$
$$\alpha(t) = t - \lfloor t \rfloor$$

Figure 2:Heat equation with discontinuous time-dependent source term, model trained with smoot

References

[1] T. Qin, **Z. Chen**, J. Jakeman and D. Xiu, Deep learning of parameterized equations with applications to uncertainty quantification, *International Journal for Uncertainty Quantification*, 2020.
[2] T. Qin, **Z. Chen**, J. Jakeman and D. Xiu, Data-driven learning of non-autonomous systems, *SIAM Journal on Scientific Computing*, in revision, 2020.