The Error Analysis of Numerical Integrators for Deep Neural Network Modeling of Differential Equations

Shunpei Terakawa, Graduate School of System and Information Science, Kobe University
Takashi Matsubara, Graduate School of Engineering Science, Osaka University
Takaharu Yaguchi, Graduate School of System and Information Science, Kobe University

Summary

Motivation: Modeling unknown dynamics

Target dynamics :

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(x)$$

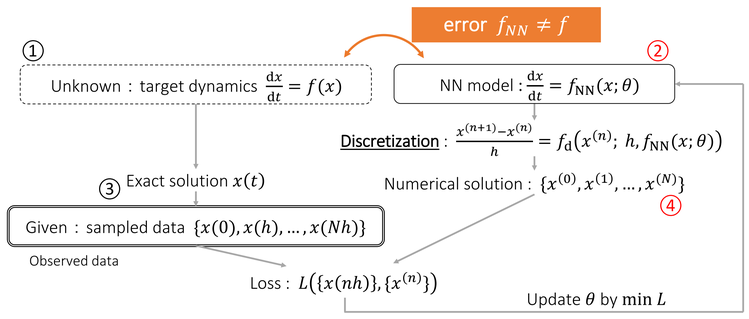
Many systems of physics are written in this form

$\{x(t_1), x(t_2), \dots, x(t_n)\}$: given data
$f(x)$: unknown

Problem :
Construct continuous model $f_{NN}(x;\theta)$ which approximates $f(x)$

Related works: NODE[Tian et al. (2018)], HNN[Greydanus et al. (2019)], etc…

Point :
absence of $\{\dot{x}(t_1), \dot{x}(t_2), \dots, \dot{x}(t_n)\}$
→ need for **discretization** of $f_{NN}(x;\theta)$ in learning process
→ causes **modeling error** $f_{NN} \neq f$



Contribution:

- Defined the modeling error as the “learnability coefficient”
- Formulated the way of learnability analysis
- Proposed the new viewpoint of designing numerical methods other than the consistency, stability, and convergence

How to use the result:

1. Roughly estimate the damping/oscillation behavior of the target dynamics (e.g. by spectrogram)
2. Select a discretization method that has good “learnability” for the behavior
3. Or, design a new method that is good in terms of “learnability”

Learnability analysis enables us to do step 2 and 3

Theory

Setup 1: The model equation

Target dynamics f is too general
-> Introduce the linear “model equation” as a benchmark problem

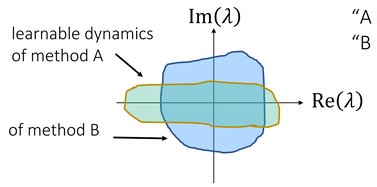
$$\text{model equation*} : \frac{\mathrm{d}x}{\mathrm{d}t} = \lambda x \quad (\lambda \in \mathbb{C})$$

*commonly used in stability analysis of numerical methods

λx represents linearized and diagonalized f
$\mathrm{Re}(\lambda)$: damping, $\mathrm{Im}(\lambda)$: oscillation

The idea of “learnability analysis”

learnable region $\{\lambda$: learnability(λ) < threshold $\}$



“A is better than B for learning damping dynamics”
“B is better than A for learning vibrating dynamics”
↑
to say like these, calc the learnability **for each discretization method** and **for each λ**

what’s “learnability”?

Setup 2: Definition of the learnability coefficient

To know the error that remains even if the learning is successful, suppose that the learned model f_{NN} is linear : $f_{NN}(x) = \alpha x \ \ (\alpha \in \mathbb{C})$

Define the relative error as “learnability coefficient” $\left|\frac{\alpha-\lambda}{\lambda}\right|$

Main result : equation of learned model

Theorem

If we learned the **model eq.** $\frac{\mathrm{d}x}{\mathrm{d}t} = \lambda x$ with the **Runge-Kutta methods**,
There exists a **linear model** $f_{\mathrm{NN}}(x) = \alpha x$ that the numerical solution of f_{NN} **matches the sampled exact solution** $x(nh) = x_0 \mathrm{e}^{n\lambda h}$. The α satisfies :

$$\boldsymbol{b}^{\mathrm{T}}(I - \alpha h A)^{-1}\mathbf{1}\alpha h - \mathrm{e}^{\lambda h} + 1 = 0, \quad ⑤$$

where $\boldsymbol{b}$ and A are constant vector and matrix that specify p-stage Runge-Kutta methods.

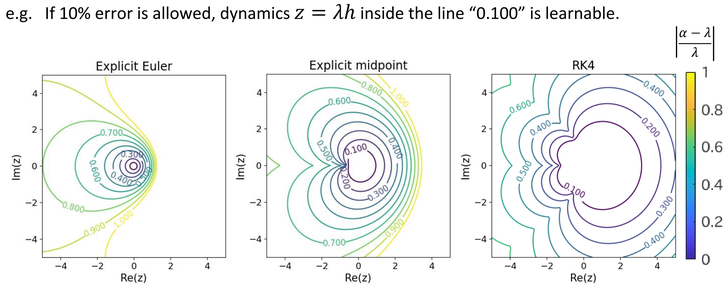
→ α and the learnability can be calculated

Example

Visualization of learnability

Learnability regions for some Runge-Kutta methods

e.g. If 10% error is allowed, dynamics $z = \lambda h$ inside the line “0.100” is learnable.

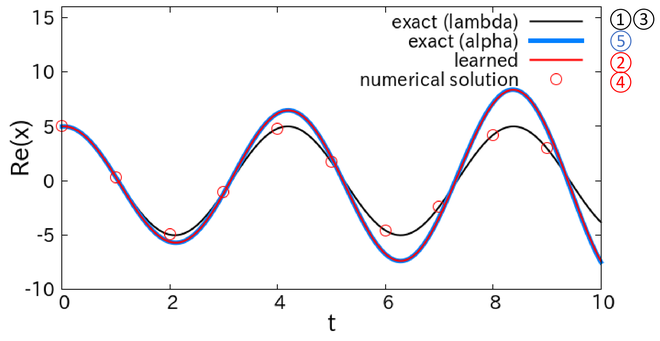


■ $\left|\frac{\alpha-\lambda}{\lambda}\right|$ is a function of $z = \lambda h$
■ h is usually fixed, so the modeling error can be controlled by the choice of discretization method

Experiment

The result of learning $\lambda = 1.5i$
with the classical 4th order Runge-Kutta ($h = 1.0$).

Learned model and prediction for $\lambda = 1.5i$



Learning is successful: ③=④
”∘”, the numerical solution of f_{NN} with the method, is on the exact solution “—”

But the learned dynamics differs from the target: ①≠②
— : the accurately integrated solution of f_{NN}
— : the target dynamics

The difference is predicted: ②=⑤
— : the learned dynamics f_{NN}
— : the theoretical prediction