Georgia Institute of Technology

Can a Neural ODE Learn a Chaotic System?

Jayjay Park, Nisha Chandramoorthy

Computational Science & Engineering Department

Informal Introduction to Dynamical System

- A system whose behavior is described by predefined rules
- Type: Discrete Time vs Continuous Time

Discrete Time Dynamical System

$$x_t = f(x_{t-1}, t)$$

- x_t = *state variable of system at time* t
- f = *function that determines the rules by which the system changes its state over time*

Lorenz: A Dynamical System that is Non-linear, Discrete-time

Lorenz System

$$\frac{dx}{dt} = \sigma(y - x)$$
$$\frac{dy}{dt} = x(\rho - z) - y$$
$$\frac{dz}{dt} = xy - \beta z$$

- Depending on the value of σ, β, ρ, lorenz system can be *fixed point*, *periodic*, and *chaotic*

[0] Edward N Lorenz. "Deterministic nonperiodic flow". In: *Journal of atmospheric sciences* 20.2 (1963), pp. 130–141.

Introducing Butterfly Attractor, Lorenz-63

Lorenz-63 is when parameters are $\sigma = 10, \beta = 8/3, \rho = 28$

- Lorenz-63 is **a chaotic system**
  - deterministic systems,
  - extreme sensitivity to initial points
  - thus behaving like a random system
- Lorenz-63 is **an ergodic system**
  - a dynamic system, whose ensemble average = time average
  - *animation*

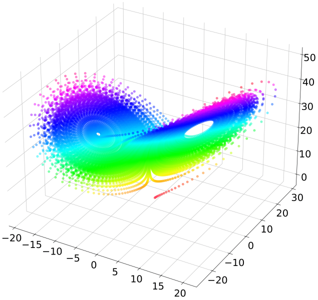

Figure: Lorenz-63

What would learning Lorenz-63 from data mean?

Learning a chaotic and ergodic system must mean that statistics are reproduced

- Learning a chaotic system would mean
  - Auto-correlation $\to 0$
  - Lyapunov Spectrum should match to True Lyapunov Spectrum $[0.9, 0, -14]$
  - Phase Plot should show **strange attractor**
- Learning an ergodic system would mean
  - Time average should converge
  - Wasserstein Distance $\to 0$

Neural ODE

Neural Ordinary Differential

$$\frac{dh(t)}{dt} = \phi_h(h(t), t, \theta) \quad s.t.\ t \in [0, T]$$

- $h(t)$ is a hidden layer which produces state at time t. Models the **dynamics**
- θ is parameter of hidden layers
- ϕ_h is time integrator of $h(t)$

[0] Ricky TQ Chen et al. "Neural ordinary differential equations". In: *Advances in neural information processing systems* 31 (2018).

Research Question 1: Can a Neural ODE learn a chaotic system?

1. **What is the learning problem of interest?**

2. **Does Neural ODE learn chaotic system?**
   - Is training loss, and train loss reasonably low?
   - How does orbit look like?
   - Does statistical properties discussed above match?
   - **Does introducing transition phase in training dataset will influence Neural ODE's learning?**

Learning Problem

- Supervised learning problem: $(x_i, \phi(x_i))$

Empirical Risk Minimization Problem: $Given\ S = \{x_i\}_{i=1}^{m},\ x \in \mathbb{R}^d,$

$$\mathbf{R}(h) = \mathop{\mathbb{E}}_{S\sim D^m} \widehat{\mathbf{R}}_s(h) = \mathop{\mathbb{E}}_{S\sim D^m} \frac{1}{m}\sum_{i=1}^{m} l(z_i, h)$$

$$MSE\_loss = l(x,h) = \|\phi_h^{\Delta t}(x) - \phi_f^{\Delta t}(x)\|^2$$

$$Neural\_ODE = \frac{d}{dt}\phi_h^t(x) = h(\phi_h^t(x)) \quad t \in \mathbb{R}^+, x \in \mathbb{R}^d, \phi_h^t(x) \in \mathbb{R}^d$$

$$True\_ODE = \frac{d}{dt}\phi_f^t(x) = f(\phi_f^t(x)) \quad t \in \mathbb{R}^+, x \in \mathbb{R}^d, \phi_f^t(x) \in \mathbb{R}^d$$

Baseline Experiment Setting

1. Architecture: 3 Layer Feed Forward Network
2. Training Algorithm: AdamW
   - Learning rate: $5e-4$
   - Number of epoch: 20000
3. Data: are generated from [0, 180] integration time.
   - Time step size: $1e-2$
   - Size of Training Data: 10000
   - Size of Test Data: 7500
4. Variable for Analysis:
   - For Training: **two transition phase**, chosen from $\{0, 3\}$ in real time

$\Rightarrow$ Two types of baseline model: MSE_0, MSE_3

Baseline Experiment Result

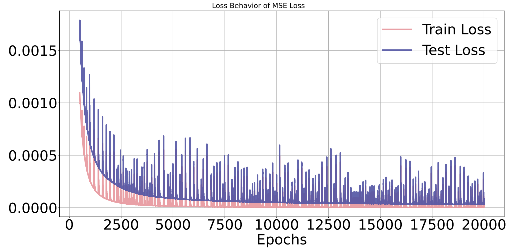

Figure: Train and Test Loss of Neural ODE without transition phase

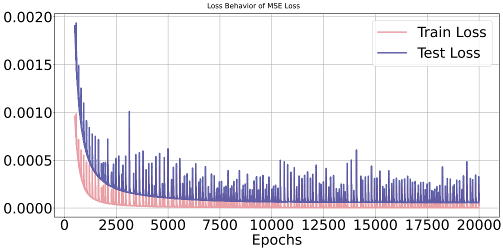

Figure: Train and Test Loss of Neural ODE with transition phase

- As expected, training loss was small
- Low test error also implies that generalization error will be low as well

Baseline Experiment Result

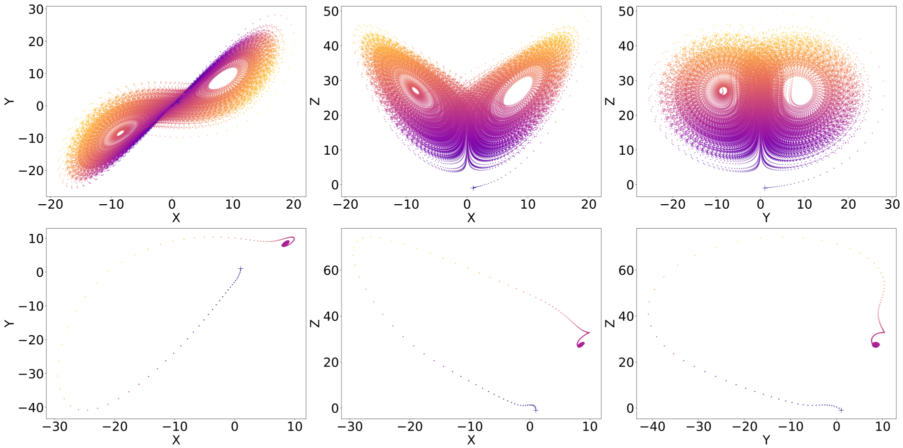

Figure: Phase Plot of True Lorenz and MSE_3's Lorenz starting from outside of attractor

Baseline Experiment Result

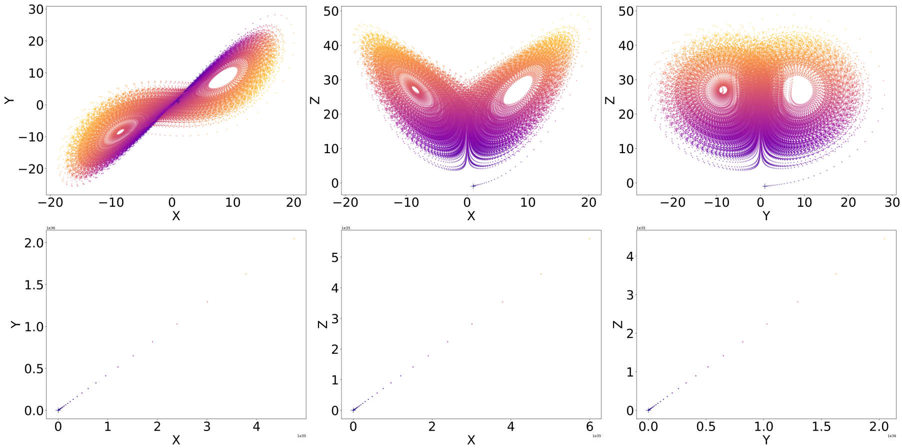

Figure: Phase Plot of True Lorenz and MSE_0's Lorenz starting from outside of attractor

Baseline Experiment Result

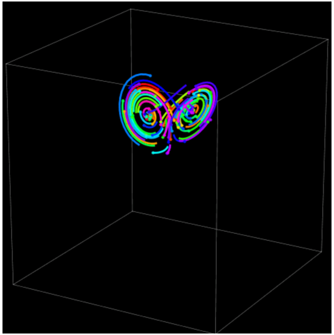

Figure: Animation of 3D Attractor

Finding

1. Transition phase in traning dataset impacts learning

	Phase Plot	LE
MSE_0	wrong	incorrect $[0.8926, -0.0336, -6.0616]$
MSE_3	wrong	incorrect $[0.9122, -0.0187, -6.1176]$

2. Neural ODE's learned dynamic is not ergodic.

$\Rightarrow$ Generalization error of Neural ODE being small does not imply that true dynamics are learned!

Research Question 2: How can we make a Neural ODE learn the true dynamics and its statistics?

1. What is our proposed algorithm?
2. Using the same metric above, can we observe that it can learn true chaotic, ergodic system?

The Proposed Algorithm

- Same supervised learning problem: $(x_i, \phi(x_i))$
- Introducing new loss function

New Empirical Risk Minimization Problem

$$\textit{Jacobian loss} = l_{new}(x,h) = \|\phi_h^{\Delta t}(x) - \phi_f^{\Delta t}(x)\|^2 + \lambda\|\nabla h(x) - \nabla f(x)\|$$

- λ is regularization parameter

New Model's Experiment Setting

1. Architecture: 3 Layer Feed Forward Network
2. Training Algorithm: AdamW
   - Learning rate: $5e-4$
   - Number of epoch: 20000
3. Data: are generated from [0, 180] integration time.
   - Time step size: $1e-2$
   - Size of Training Data: 10000
   - Size of Test Data: 7500
4. Variable for Analysis:
   - For training, **Transition phase**: chosen from $\{0, 3\}$ in real time

$\Rightarrow$ **Two types of new model: JAC_0, JAC_3**

Summary of Two Experiments

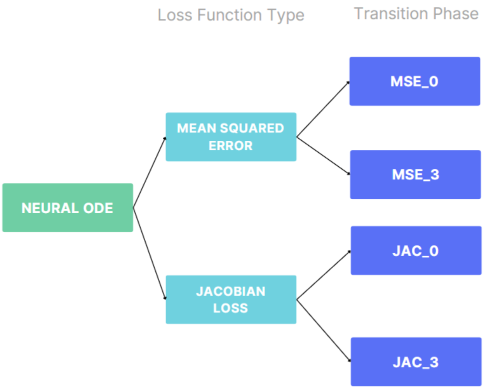

Figure: 4 Models

- "0" means transition phase is included
- "3" means transition phase of 300 data points is excluded
- **But what difference does it make?**

Experiment Result 1: Loss Behavior

	Train Loss (Jac or MSE)	Test Loss (MSE)
MSE_0	$1.1158e-05$	$3.0489e-05$
MSE_3	$9.6705e-05$	0.0001
JAC_0	1.1301	$\mathbf{9.6022e-06}$
JAC_3	1.0507	$2.6281e-05$

Table: Loss

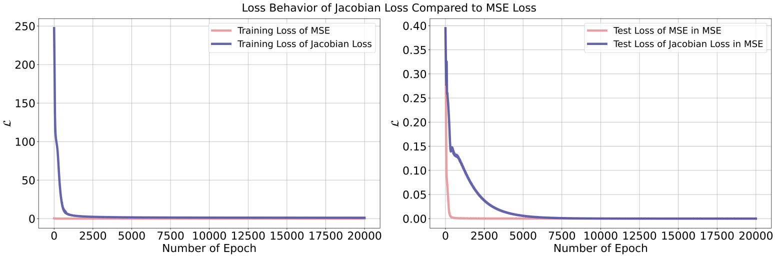

Figure: Train & Test Loss of MSE and JAC

Experiment Result 2: Orbit

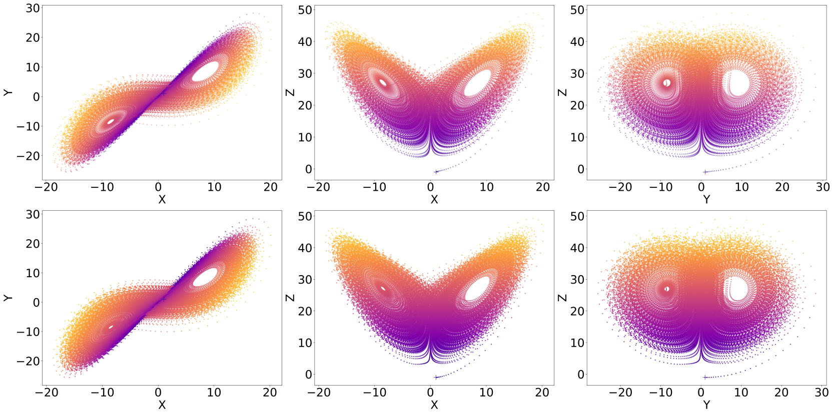

Figure: Phase plot from JAC_0 solution

Experiment Result 3-1: Wasserstein Distance

Model	from attractor	out of attractor
MSE_0	$[0.2211, 0.2188, 0.2597]$	trajectory explodes
MSE_3	$[4.9432, 5.1924, 3.7964]$	$[10.2379, 10.8456, 7.8666]$
JAC_0	$[\mathbf{0.2649}, \mathbf{0.2863}, \mathbf{0.0934}]$	$[\mathbf{1.0547}, \mathbf{1.0669}, \mathbf{0.0991}]$
JAC_3	$[0.5337, 0.5399, 0.1708]$	$[1.0872, 1.1359, 0.3524]$

Table: Wasserstein Distance

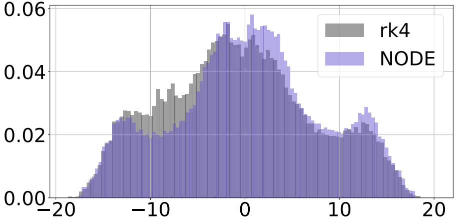

Figure: Distribution for MSE_0's x component

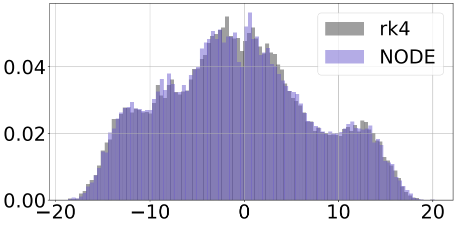

Figure: Distribution for JAC_0's x component

Experiment Result 3-2: Time Average

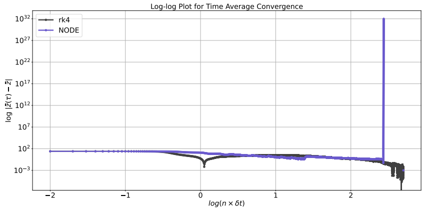

Figure: MSE_0, Init_Point = [1.0, 1.0, -1.0]

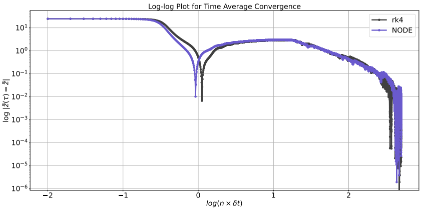

Figure: JAC_0, Init_Point = [1.0, 1.0, -1.0]

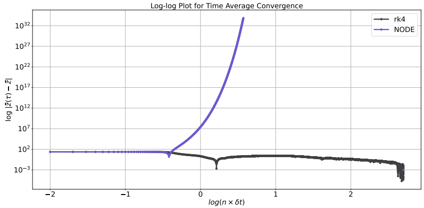

Figure: MSE_0, Init_Point = [1.0, 0., 0.]

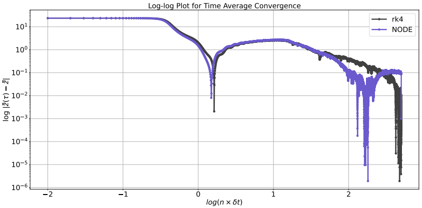

Figure: JAC_0, Init_Point = [1.0, 0., 0.]

Experiment Result 3-3: Lyapunov Exponent

	Lyapunov Exponent	Norm Difference
True LE	$[0.9, 0, -14]$	
MSE_0	$[0.8926, -0.0336, -6.0616]$	8.4715
MSE_3	$[0.9122, -0.0187, -6.1176]$	8.4155
JAC_0	$[0.9022, -0.0024, -14.4803]$	$\mathbf{0.0655}$
JAC_3	$[0.8493, 0.099, -14.5299]$	0.0973

Table: Lyapunov Exponent

Experiment Result 3-4: Auto-Correlation

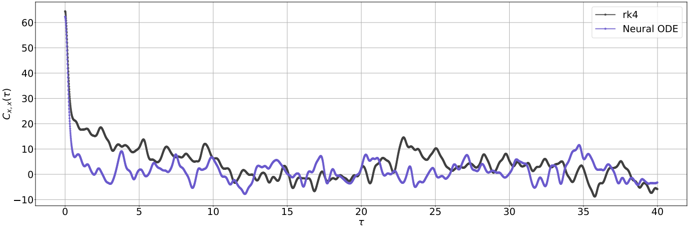

Figure: MSE_0, Init_Point = $[1.0, 1.0, -1.0]$

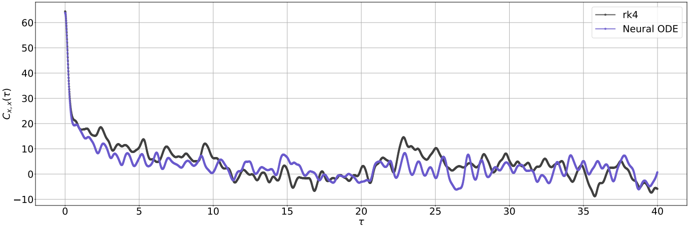

Figure: JAC_0, Init_Point = $[1.0, 1.0, -1.0]$

Finding

- Adding Jacobian to the loss for Neural ODE learns the correct dynamics for lorenz-63 and its statistics! $\Rightarrow$ ergodic, and chaotic dynamics
  - Better simulated auto-correlation
  - Computes correct Lyapunov Spectrum
  - Reproduces correct phase plot
  - Time average converges
  - Simulated dataset shows similar distribution

Future Work

- For Lyapunov Exponents, it makes sense that adding jacobian to loss will lead to better estimation of LEs
- But in general, does this work for any dynamical system? Why?
- **Must redefine generalization error to reflect true learning of ergodic dynamics**

Thank you for coming! Any Questions?