Reservoir Computing on Non-Linear Schrödinger Equation

A. Hasmi[1]

[1]PhD Student of Mathematics
Khalifa University

21 February 2022

Outline

1. Theory
2. Result
    - Non-Linear Schrödinger

Data Driven Method for Partial Differential Equation

Frame subtitles are optional. Use upper- or lowercase letters.

- First Principle Methods: Discretize PDEs using Numerical Schemes: Finite Difference, Finite Element, Finite Volume, Spectral Methods, etc
- Data Driven Method: Incorporate Observation/Data to simulate PDEs: Physics Informed Neural Network,
- Pros/Reasoning Using DDM:
    - Learning Hidden Physics from Data (Turbulence, Wall Model)
    - Reducing Computational Cost (Reduced Order Method,)
    - Data Assimilation with observation data[3]

Reservoir Computing

- Recurrent Neural Network but only train the output layer [2]
- Fast training as only output parameters need to optimized [1, 2]
- Does not suffer from vanishing gradient problem

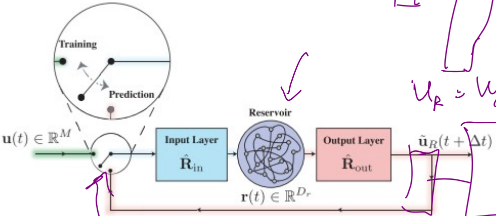
input

output

R_1

R_2

R_3



$u_R = W_{out} r$

Reservoir Computing for Spatiotemporal Problem

Reservoir   Reservoir

X_1 X_2 X_3 X_4 X_5 X_6 X_7 X_8

- By locality of interaction between grids, we can split the spatial domain into several reservoirs and train them independently
- internal states updates:

input

$$r_{i+1}^{(k)} = \tanh\left(W^{(k)} r_i^{(k)} + W_{in}^{(k)} v_i^{(k)}\right)$$

- output readout

$$y_i^{(k)} = W_{out}^{(k)} \hat{r}_i^{(k)} \qquad \hat{r}_{ij}^{(k)} = \begin{cases} r_{ij}^{(k)} & j \text{ is odd} \\ \left(r_{ij}^{(k)}\right)^2 & j \text{ is even} \end{cases}$$

$|r_{ij}|^2 r_{ij}$

- Training Objectives

$$E^{(k)} = \min_{W_{out}^{k}} \sum_{i=1}^{N_{train}} \left\| W_{out}^{(k)} \hat{r}_i^{(k)} - s_i^{(k)} \right\|_2^2 + \beta \left\| W_{out}^{(k)} \right\|^2$$

input   input

x_{grid}

Prediction Steps

State space

1. Compute $y_i^{(k)} = W_{out}^k \hat{r}_i^{(k)}$
2. Construct $\tilde{v}_i^{(k)}$ based on $y_i^{(k-1)}, y_i^{(k)} y_i^{(k+1)}$
3. Update the reservoir states: $r_{i+1}^{(k)} = \tanh\left(W^{(k)} r_i^{(k)} + W_{in}^{(k)} \tilde{v}_i^{(k)}\right)$

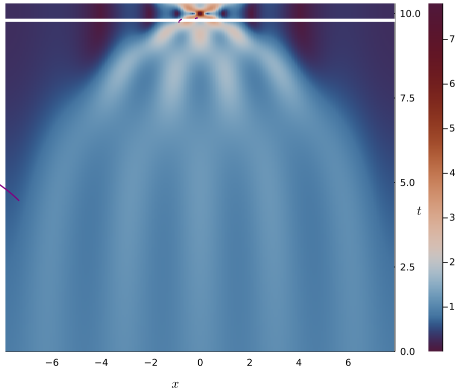
Outline

Non-Linear Schrödinger

- Equation given by:

$$i\frac{\partial \psi}{\partial \tau} + \frac{1}{2}\frac{\partial^2 \psi}{\partial \xi^2} + |\psi|^2 \psi = 0$$

- Can be used to modeled rogue wave
- The figure shows training and testing data for NLS. The limit of training data is given by whiteline



Simulation Hyperparameter

Parameter	Explanation	Value
ρ	Spectral Radius of W	0.6
σ	maximum elementwise magnitude of W_{in}	0.1
β	Regularization Parameter	10^{-4}
M	Number of reservoirs	32
l	Number of overlap grid	6

NLT1

$$\hat{r}_{ij}^{(k)} = \begin{cases} r_{ij}^{(k)} & j \text{ is odd} \\ \left(r_{ij}^{(k)}\right)^2 & j \text{ is even} \end{cases}$$

NLT2

$$\hat{r}_{ij}^{(k)} = \begin{cases} r_{ij}^{(k)} & j \text{ is odd} \\ \left|r_{ij}^{(k)}\right|^2 r_{ij}^{(k)} & j \text{ is even} \end{cases}$$

Assessing the Result

To measure the performance, we use H^1 norm (Sobolev space norm). Let $u \in H^1$, then the norm is defined as:

$$\|u\|_{H^1} = \left[\int |u|^2 \, dx\right]^{\frac{1}{2}} + \left[\int |du|^2 \, dx\right]^{\frac{1}{2}}$$

the error is calculated at $t = 10$ (the expected peak), with error defined as:

$du = F^{-1}\{ik^2 F(u)\}$

$$e = \frac{\left\|\hat{\psi} - \psi\right\|_{H^1}}{\|\psi\|_{H^1}}$$

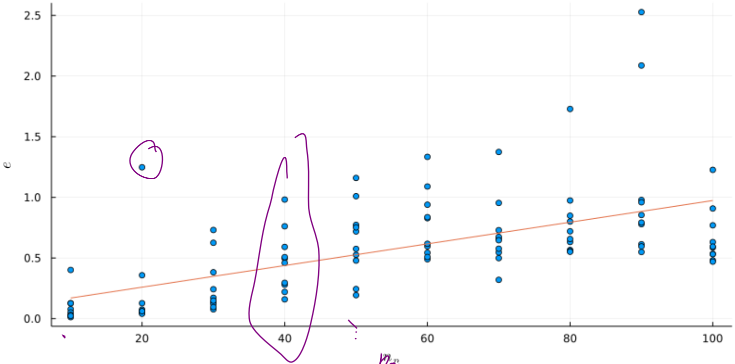
$u =$

Result: Prediction Length

Simulation using NLT1, 200 reservoir nodes with varying start of prediction phase. 10 realizations for each case.

Ω

$\frac{n_p \Delta t}{\Omega}$

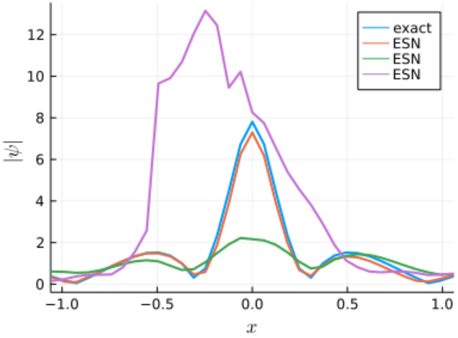


t

$10-10\Delta t$ $10-2\Delta t$ $10-\Delta t$ 10

Typical Result

there are typical success simulations however, the simulations can be failed.



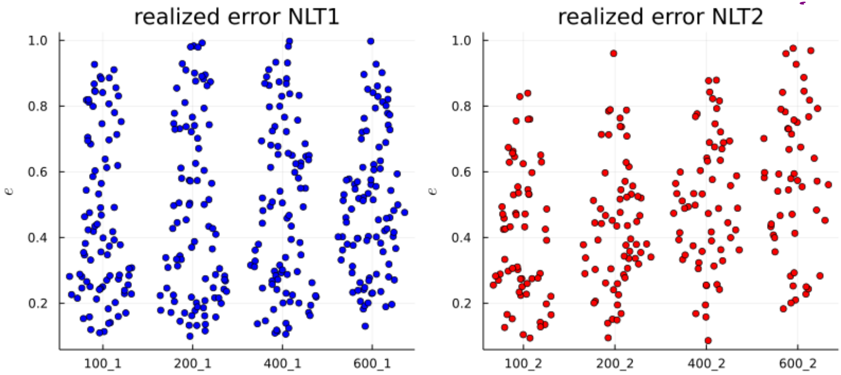
e = 0.5

ensemble.

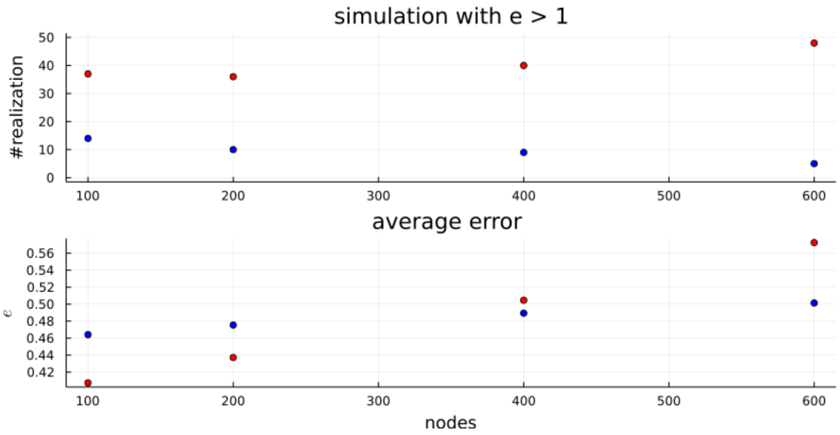
Fata 2

Result:Effect of Reservoir Nodes

$n_p = 40$. nodes: 100, 200,400, 600 with both NLT1 and NLT2 .The simulations are repeated for 100 realization,



Result: Effect of Reservoir Nodes



Summary

- Simulation Quality detoriates farther away from prediction
- NLT1 gives more robust result compared to NLT1
- Only 100 or 200 nodes are enough for prediction
- Outlook
  - Optimization of hyperparameter?
  - Can we improve the robustness of the result?

For Further Reading I

Ashesh Chattopadhyay, Pedram Hassanzadeh, and Devika Subramanian.
Data-driven predictions of a multiscale Lorenz 96 chaotic system using machine-learning methods: Reservoir computing, artificial neural network, and long short-term memory network.
Nonlinear Processes in Geophysics, 27(3):373–389, 2020.

Mantas Lukoševičius and Herbert Jaeger.
Reservoir computing approaches to recurrent neural network training.
Computer Science Review, 3(3):127–149, 2009.

Jaideep Pathak, Alexander Wikner, Rebeckah Fussell, Sarthak Chandra, Brian R. Hunt, Michelle Girvan, and Edward Ott.
Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model.
Chaos, 28(4), 2018.

For Further Reading II