Analyzing robustness of models of chaotic dynamical systems learned from data with Echo state networks

M.S. Thesis Defense

Mohamed Abdelrahman

Rice University

July 29, 2019

Acknowledgement

Committee Members:

Prof. Devika Subramanian (Committee chair)
Prof. Robert S. Cartwright, Jr. (Committee member)
Prof. Krishna V. Palem (Committee member)
Dr. Pedram Hassanzadeh (Committee member)

Contents

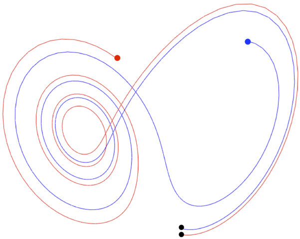
Contents

Background

- Dynamical system: states evolving over time

Background

- Dynamical system: states evolving over time
- Chaotic dynamical system: two trajectories that start with initial states that are infinitesimally apart will diverge over time [Kuc13]. The distance between the two trajectories at time (t) is: $D_t = e^{\lambda t} D_0$, λ is the maximum Lyapunov exponent of the system.



Problem definition

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

- Modeling the behavior of chaotic systems, set of coupled ODEs or PDEs are formulated and solved using high performance computing systems.

Problem definition

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

- Modeling the behavior of chaotic systems, set of coupled ODEs or PDEs are formulated and solved using high performance computing systems.
- Given a limited time series of past measurements can we predict the future state of the chaotic system — i.e.,the short term evolution of the chaotic dynamical system?

Problem definition

$$\frac{dx}{dt} = \sigma(y - x)$$

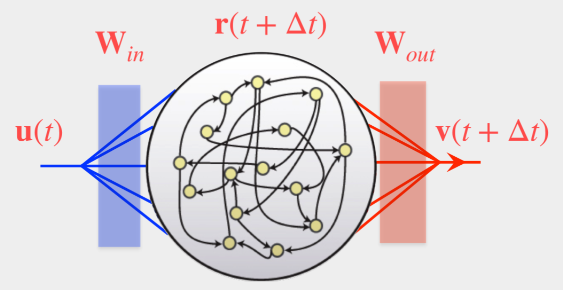
$$\frac{dy}{dt} = x(\rho - z) - y$$

$$\frac{dz}{dt} = xy - \beta z$$

- Modeling the behavior of chaotic systems, set of coupled ODEs or PDEs are formulated and solved using high performance computing systems.
- Given a limited time series of past measurements can we predict the future state of the chaotic system — i.e.,the short term evolution of the chaotic dynamical system?
- Can we learn something about the long-term dynamics of the chaotic system?

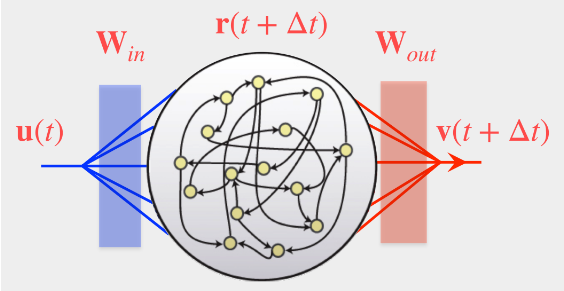
Contents

ESN structure [Pat18]



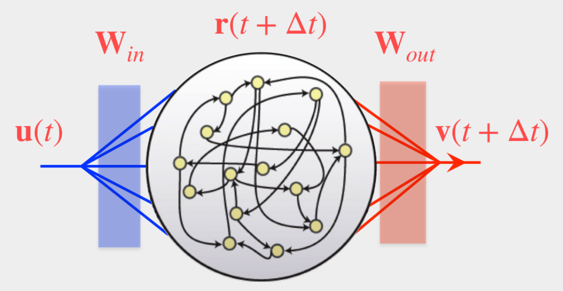
- The reservoir $r(t)$ in the network is of size D_r nodes

ESN structure [Pat18]



- The reservoir $r(t)$ in the network is of size D_r nodes
- Each node i has multiple inputs and outputs and a scalar state denoted by $r_i(t)$

ESN structure [Pat18]



- The reservoir $r(t)$ in the network is of size D_r nodes
- Each node i has multiple inputs and outputs and a scalar state denoted by $r_i(t)$
- The weighted connections between the nodes can be represented by an adjacency matrix A

ESN parameters [Pat18]

- Spectral radius ρ defines the largest absolute eigenvalue of the matrix A. It represents how much the current state is determined by previous inputs vs. the initial state.

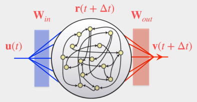
ESN parameters [Pat18]

- Spectral radius ρ defines the largest absolute eigenvalue of the matrix A. It represents how much the current state is determined by previous inputs vs. the initial state.
- Density d_A is the fraction of nonzero entries in the connection matrix A. This controls the level of sparsity in the matrix. The lower the density the faster the computation times for large networks.

ESN parameters [Pat18]

- Spectral radius ρ defines the largest absolute eigenvalue of the matrix A. It represents how much the current state is determined by previous inputs vs. the initial state.
- Density d_A is the fraction of nonzero entries in the connection matrix A. This controls the level of sparsity in the matrix. The lower the density the faster the computation times for large networks.
- Reservoir Size D_r is the number of neurons in the reservoir.

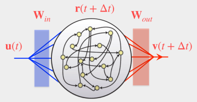
ESN training [Pat18]



- An input is coupled to the reservoir network through a fixed, randomly generated input matrix.

$$r(t+\Delta t) = tanh(A\ r(t) + W_{in}\ u(t)), -T \leq t \leq 0$$

ESN training [Pat18]



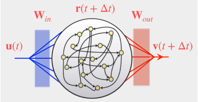
- An input is coupled to the reservoir network through a fixed, randomly generated input matrix.

$$r(t+\Delta t) = tanh(A\ r(t) + W_{in}\ u(t)), -T \leq t \leq 0$$

- Nonlinear transformation to capture nonlinearity in chaotic systems. Change odd columns of $\tilde{r}$ [Cha19] as:

$$\tilde{r}_{i,j} = r_{i,j-1}\ r_{i,j-2}$$

ESN training [Pat18]



- An input is coupled to the reservoir network through a fixed, randomly generated input matrix.

$$r(t+\Delta t) = tanh(A\ r(t) + W_{in}\ u(t)), -T \leq t \leq 0$$

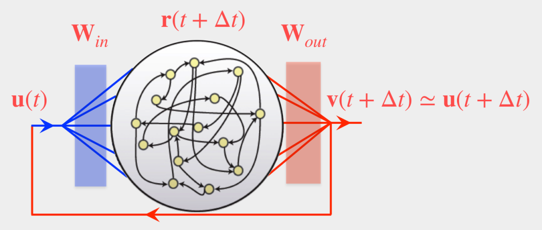
- Nonlinear transformation to capture nonlinearity in chaotic systems. Change odd columns of $\tilde{r}$ [Cha19] as:

$$\tilde{r}_{i,j} = r_{i,j-1}\ r_{i,j-2}$$

- Only output weights are learned:

$$W_{out} = argmin_{W_{out}} ||W_{out}\ \tilde{r}(t) - u(t)|| + \alpha ||W_{out}||$$

ESN prediction [Pat18]



- During prediction $t > 0$, W_{out} advances $v(t)$ in time while $r(t)$ gets updated using the predicted u(t) by the following equations:

$$v(t+\Delta t) = W_{out}\ \tilde{r}(t+\Delta t)$$
$$u(t+\Delta t) = v(t+\Delta t)$$

Lorenz63 PDEs

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

- Lorenz [Lor63] exhibits chaotic behavior of the system when $\sigma = 10.0$, $\rho = 28.0$ and $\beta = \frac{8}{3}$.

ESN parameters to study Lorenz63 [Pat17]

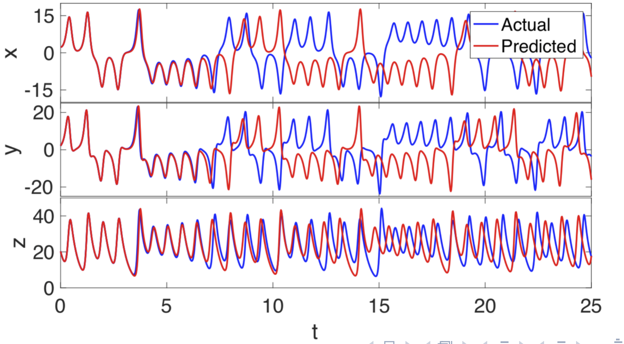
- ESN was trained on $(u_{-T}, \ldots, u_0)$ where each $u = (x, y, z)$ and predicted $(u_1, \ldots, u_t)$

ESN parameters to study Lorenz63 [Pat17]

- ESN was trained on $(u_{-T}, ..., u_0)$ where each $u = (x, y, z)$ and predicted $(u_1, ..., u_t)$
- ESN main parameters: $D_r = 300, d_A = 0.02, \rho = 1.2, T = 100$

ESN parameters to study Lorenz63 [Pat17]

- ESN was trained on $(u_{-T}, ..., u_0)$ where each $u = (x, y, z)$ and predicted $(u_1, ..., u_t)$
- ESN main parameters: $D_r = 300, d_A = 0.02, \rho = 1.2, T = 100$
- Results: Accurate predictions ≈ 7 MTU



Lorenz96 PDEs

$$\frac{dX_k}{dt} = X_{k-1}(X_{k+1} - X_{k-2}) + F - \frac{hc}{b}\sum_j Y_{j,k}$$

$$\frac{dY_{j,k}}{dt} = -cbY_{j+1,k}(Y_{j+2,k} - Y_{j-1,k}) - cY_{j,k} + \frac{hc}{b}X_k - \frac{he}{d}\sum_i Z_{i,j,k}$$

$$\frac{dZ_{i,j,k}}{dt} = edZ_{i-1,j,k}(Z_{i+1,j,k} - Z_{i-2,j,k}) - eZ_{i,j,k} + \frac{he}{d}Y_{j,k}$$

- Proposed [Lor96] to model multi-scale chaotic variability of the climate system.

Lorenz96 PDEs

$$\frac{dX_k}{dt} = X_{k-1}(X_{k+1} - X_{k-2}) + F - \frac{hc}{b}\sum_j Y_{j,k}$$

$$\frac{dY_{j,k}}{dt} = -cbY_{j+1,k}(Y_{j+2,k} - Y_{j-1,k}) - cY_{j,k} + \frac{hc}{b}X_k - \frac{he}{d}\sum_i Z_{i,j,k}$$

$$\frac{dZ_{i,j,k}}{dt} = edZ_{i-1,j,k}(Z_{i+1,j,k} - Z_{i-2,j,k}) - eZ_{i,j,k} + \frac{he}{d}Y_{j,k}$$

- Proposed [Lor96] to model multi-scale chaotic variability of the climate system.
- $i, j, k = 1:8$, X, Y, Z have $8, 64, 512$ elements. X has large amplitudes and slow variability; Y and Z have small amplitudes and high variability.

ESN parameters to study Lorenz96 [Cha19]

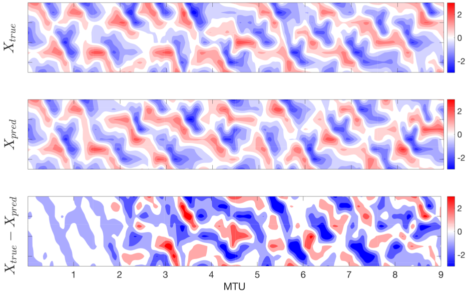
- Only $X(t)$ with its 8 elements were considered in this work for training and prediction.

ESN parameters to study Lorenz96 [Cha19]

- Only $X(t)$ with its 8 elements were considered in this work for training and prediction.
- ESN main parameters:
  $D_r = 5000, d_A = 0.0006, \rho = 0.1, T = 500000$

ESN parameters to study Lorenz96 [Cha19]

- Only $X(t)$ with its 8 elements were considered in this work for training and prediction.
- ESN main parameters:
  $D_r = 5000, d_A = 0.0006, \rho = 0.1, T = 500000$
- Results: Accurate predictions $\approx$ 2 MTU



Contents

Motivating questions

- How robust is the ESN's prediction to random initialization of input weights W_{in} and reservoir weights A at the start of the training phase?

Motivating questions

- How robust is the ESN's prediction to random initialization of input weights W_{in} and reservoir weights A at the start of the training phase?
- How robust is the ESN's prediction to the choice of initial condition (IC) to start the training sequence?

Motivating questions

- How robust is the ESN's prediction to random initialization of input weights W_{in} and reservoir weights A at the start of the training phase?
- How robust is the ESN's prediction to the choice of initial condition (IC) to start the training sequence?
- What properties of the underlying dynamical system are learned by the ESN?

Running Environment

- All computational resources, on Wrangler cluster at Texas Advanced Computing Center (TACC), were provided by the NSF XSEDE allocation ATM170020. The main technical specifications of the Wrangler cluster at TACC:

Component	Technology	Size
Compute Nodes	dual CPU 12 Core	96 nodes/2304 cores
Memory	Distributed DDR4	12.2 TB

Running Environment

- All computational resources, on Wrangler cluster at Texas Advanced Computing Center (TACC), were provided by the NSF XSEDE allocation ATM170020. The main technical specifications of the Wrangler cluster at TACC:

Component	Technology	Size
Compute Nodes	dual CPU 12 Core	96 nodes/2304 cores
Memory	Distributed DDR4	12.2 TB

- All of the code snippets, numerical solvers, datasets, and plots can be accessed at: `https://github.com/Mohamed-Abdelrahman/RCESN_spatio_temporal`

Lorenz63 Dataset

The Lorenz63 dataset was generated using a numerical solver with the following main parameters:

Parameter	Value
Dataset	x, y, z
Time step (ΔT)	0.01
Number of time steps (T)	5,000,000
System parameters (σ, ρ, β)	(10, 28, 2.667)

ESN Parameters

In our experiments we used a training length of 40,000 time steps and tested the network on a prediction length of 2,000 time steps:

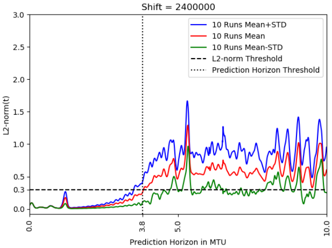
Parameter	Value
D_r: Reservoir Size	1500
ρ: Spectral Radius	0.9
d_A: Density	0.002

*Predictions w.r.t. random initialization of fixed weights

To test the robustness of ESN in response to the random initialization of W_{in} and A, we executed 10 runs with different initialization seeds.

The L2-norm $e(t)$ is calculated as:

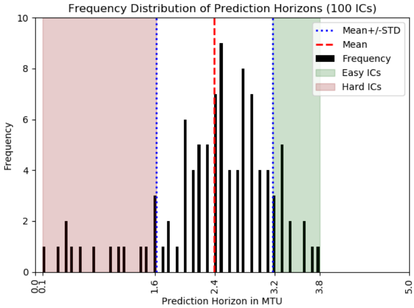
$$e(t) = \frac{||X_{true}(t) - X_{pred}(t)||}{[||X_{true}(t)||]}$$



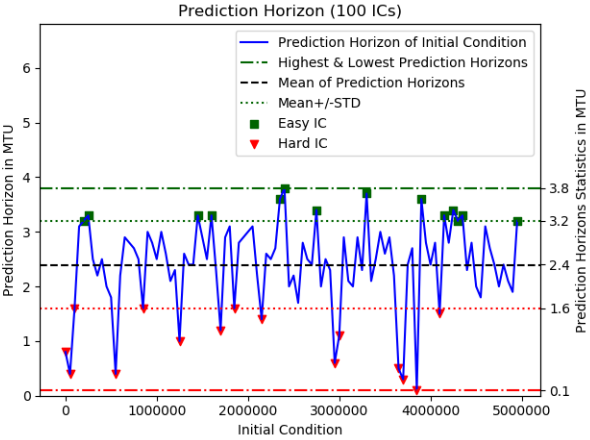
**Predictions w.r.t. different initial conditions

To test the robustness of ESN in response to the position where we start the training sequence, we ran the network to start from 100 different ICs governed by:

$$IC(i+1) = IC(i) + 50,000, IC(0) = 0$$



Easy vs. Hard ICs

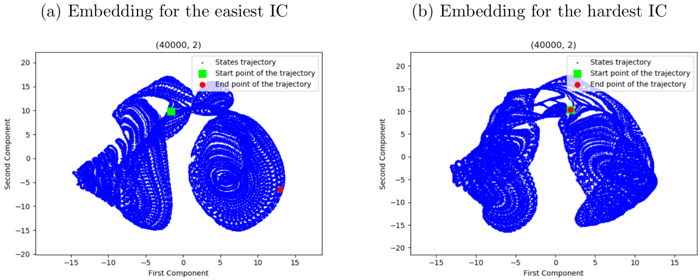


Why do models learned from different ICs have different prediction horizons?

Analyzing the evolution of reservoir states $r(t)$ in R^{1500} by embedding it using UMAP [McI18] in two dimensions.

UMAP Parameter	Value
n_components: The reduced dimension space	2
metric: How distance is computed	corr., cos., **eucl.**
n_neighbors: Points to learn the data structure	[5:**35**:5]
min_dist: How tight to pack points together	0.1,0.2,**0.3**

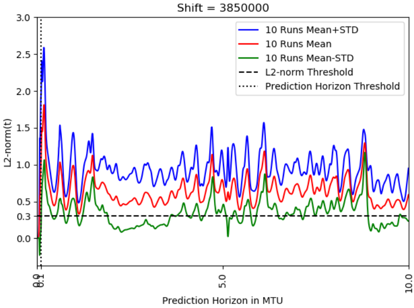
UMAP embedding of reservoir states in 2D



How does the start point of the trajectory affect the prediction horizon in (b)?

Why hard ICs have lower prediction horizons?

The trajectory started at the border and faced difficulties in the beginning finding one of the two lobes.



***Capturing the dynamical modes of the system

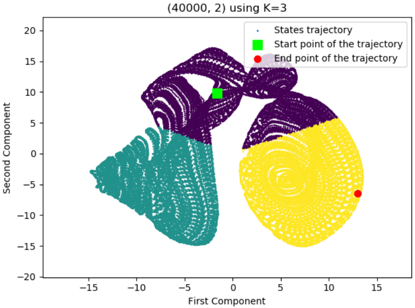
Using embedded matrix of (40000,2) we used KMeans for $K = 2:20$. The number of dynamical modes of the systems corresponds to the the K with the highest Silhouette score. Assuming $d(i,j)$ is the distance between points i and j in cluster C_i, the Silhouette score $S(i)$ for a point i is defined as:

$$S(i) = \frac{Inter(i) - Intra(i)}{max(Intra(i), Inter(i))},$$

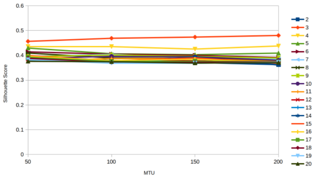
$$Intra(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i,j),$$

$$Inter(i) = min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i,j)$$

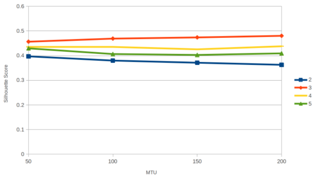
Captured modes of Lorenz63



(a) All clusters



(b) Clusters 2:5



Notice the transition states between the two lobes which is considered a cluster by KMeans.

Lorenz96 Dataset

The Lorenz96 dataset was generated using a 4th-order Runge-Kutta solver with the following main parameters:

Parameter	Value
Dataset	8 components of X
Time step (ΔT)	0.005
Number of time steps (T)	1,000,000
System parameters (F, c, b, h, e, d)	(20, 10, 10, 1, 10, 10)

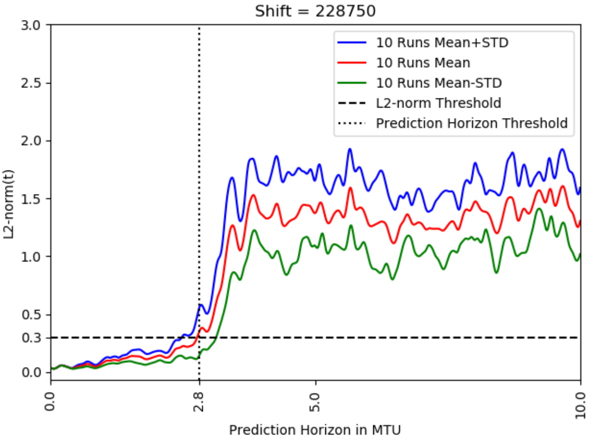
Rice University School of Engineering Department of Computer Science

ESN Parameters

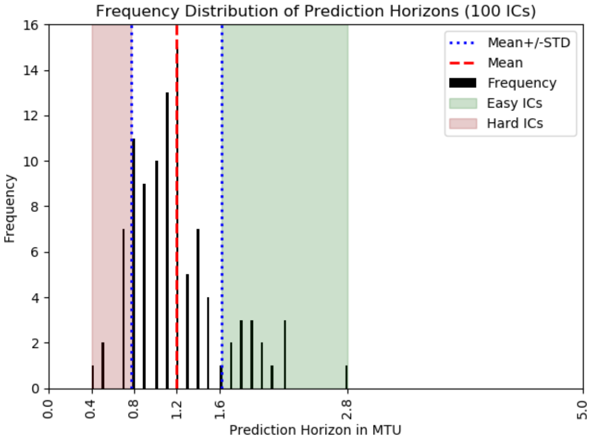
In our experiments we used a training length of 100,000 time steps and tested the network on a prediction length of 2,000 time steps:

Parameter	Value
D_r: Reservoir Size	5000
ρ: Spectral Radius	0.1
d_A: Density	0.0006

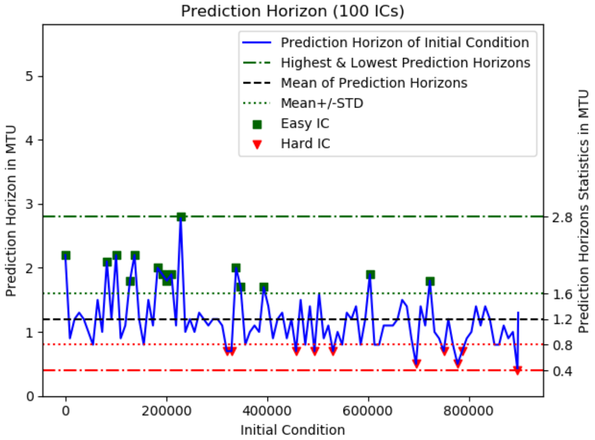
*Predictions w.r.t. random initialization of fixed weights



**Predictions w.r.t. different initial conditions

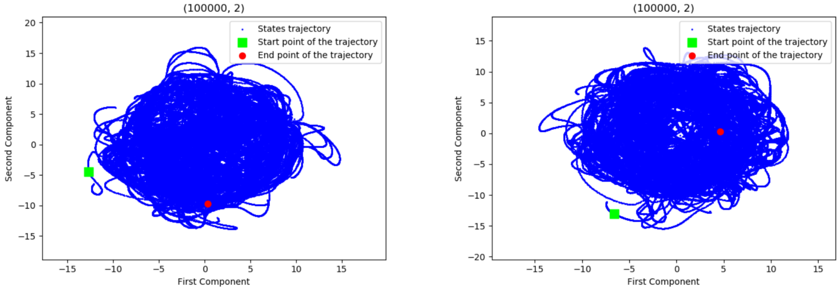


Easy vs. Hard ICs



UMAP embedding of reservoir states in 2D

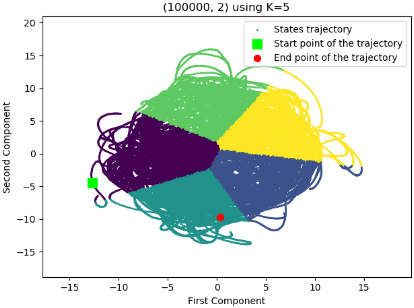
Analyzing the evolution of reservoir states $r(t)$ in R^{5000} by embedding it using UMAP [McI18] in two dimensions.



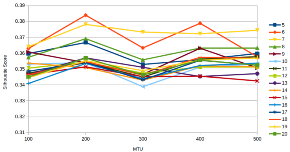
(a) Embedding for the easiest IC (b) Embedding for the hardest IC

***Capturing the dynamical modes of the system

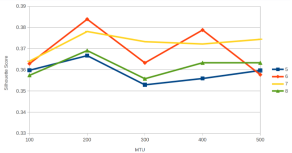
Using embedded matrix of (100000,2) we used KMeans for $K = 5:20$. The number of dynamical modes of the systems corresponds to the the K with the highest Silhouette score.



(a) All clusters



(b) Clusters 5:8



Rice University School of Engineering

Contents

Conclusion

- ESN is sensitive to the initial state of the training sequence, there is considerable variation in prediction horizon from 0.1 MTU to 3.8 MTU in Lorenz63 and from 0.4 MTU to 2.8 MTU in Lorenz96.

Conclusion

- ESN is sensitive to the initial state of the training sequence, there is considerable variation in prediction horizon from 0.1 MTU to 3.8 MTU in Lorenz63 and from 0.4 MTU to 2.8 MTU in Lorenz96.
- ESN is sensitive to variations in the initialization of (random) input weights and (random) reservoir weights at the start of the training phase that produced varying prediction horizons for the very same training sequence;
  $\pm$ 0.8 MTU in Lorenz63 and $\pm$ 0.4 MTU in Lorenz96.

Conclusion

- ESN is sensitive to the initial state of the training sequence, there is considerable variation in prediction horizon from 0.1 MTU to 3.8 MTU in Lorenz63 and from 0.4 MTU to 2.8 MTU in Lorenz96.
- ESN is sensitive to variations in the initialization of (random) input weights and (random) reservoir weights at the start of the training phase that produced varying prediction horizons for the very same training sequence;
  $\pm$ 0.8 MTU in Lorenz63 and $\pm$ 0.4 MTU in Lorenz96.
- We were able to infer the number of dynamical modes from non-linear clustering of the reservoir states.

Contents

Challenges and Open Problems

- Extract a pattern of the prediction horizons according to the initial conditions, not only knowing the positions of the Easy and Hard initial conditions, but also to locate any hidden repeated patterns of initial conditions (positions) that are expected to have similar prediction horizon.
  More ICs, different sampling, defining Easy and Hard regions

Challenges and Open Problems

- Extract a pattern of the prediction horizons according to the initial conditions, not only knowing the positions of the Easy and Hard initial conditions, but also to locate any hidden repeated patterns of initial conditions (positions) that are expected to have similar prediction horizon.
  More ICs, different sampling, defining Easy and Hard regions
- Capturing the modes of chaotic systems need more visualization capabilities especially for Lorenz96.
  `http://datashader.org/`, UMAP embedding in higher dimensions

Challenges and Open Problems

- Extract a pattern of the prediction horizons according to the initial conditions, not only knowing the positions of the Easy and Hard initial conditions, but also to locate any hidden repeated patterns of initial conditions (positions) that are expected to have similar prediction horizon.
  More ICs, different sampling, defining Easy and Hard regions
- Capturing the modes of chaotic systems need more visualization capabilities especially for Lorenz96.
  `http://datashader.org/`, UMAP embedding in higher dimensions
- More robust protocols for training ESNs which account for variations in W_{in} and A initializations.

References I

Ashesh Chattopadhyay. "Data-driven prediction of a multi-scale Lorenz 96 chaotic system using a hierarchy of deep learning methods: Reservoir computing, ANN, and RNN-LSTM". In: (2019).

Adam Kucharski. *The nature of chaos*. theconversation.com, 2013. URL: https://theconversation.com/forecasting-the-chaos-of-tornadoes-14573.

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

Edward N Lorenz. "Predictability: A problem partly solved". In: *Proc. Seminar on predictability*. Vol. 1. 1. 1996.

References II

- L. McInnes. "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction". In: *ArXiv e-prints* (Feb. 2018). arXiv: 1802.03426 [stat.ML].
- Jaideep Pathak. *MACHINE LEARNING FOR ANALYSIS OF HIGH-DIMENSIONAL CHAOTIC SPATIOTEMPORAL DYNAMICAL SYSTEMS*. Princeton Plasma Physics Laboratory Theory Seminar. 2018.
- Jaideep Pathak. "Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 27.12 (2017), p. 121102.

Thanks!