MEMORY IN ECHO STATE NETWORKS AND THE CONTROLLABILITY MATRIX RANK

Brian Whiteaker and Peter Gerstoft

Scripps Institution of Oceanography, University of California at San Diego, La Jolla, CA 92093-0238

ABSTRACT

Echo State Networks (ESNs) are a variant of recurrent neural networks (RNNs). ESNs perform as nonlinear fading memory filters and excel in prediction of "chaotic" signals. Predictions are made using a forced nonlinear dynamical system called the "reservoir" which incorporates past information into new states. The length of memory is critical to a task effective ESN. We examine the rank behavior of minimal task-effective ESNs predicting the chaotic Lorenz 1963 system for single and multi-variable input/output. Relationships are observed between the rank of the controllability matrix, memory length, and attractor features. We find that reservoir memory varies dependent on input forcing and location in state space. Knowledge of controllability matrix rank indicates a signal specific range for memory length. This variability corresponds to the reservoir varying between stable and unstable. The controllability matrix rank can facilitate efficient use of data and ESN construction.

Index Terms— recurrent neural network, echo state property, controllability matrix, nonlinear dynamical system

1. INTRODUCTION

The ESN provides a simple solution to issues endemic to RNNs[1]. The nonlinear dynamical system, called "reservoir", provides dynamics generated with the influence of previous states. These dynamics are used to make output predictions. Since the advent of ESNs[2], much effort has focused on measuring memory capacity or its relation to the "edge of chaos"[3, 4, 5, 6, 7, 8, 9]. Near this edge greatest memory and best performance occur. Another major effort is construction of reservoirs capable of delivering desirable dynamics and memory relevant to the target signal [10, 11, 12, 13, 14].

Largely, memory is treated as fixed after the model is initialized and trained. Work by Boedecker *et al.* [15] focused on relations between information transfer, information storage, and criticality. He noted that a data specific minimal model is desirable and not all of an ESNs memory capacity is utilized. Some past information is irrelevant to current outputs. Resulting wasteful matrix operations are computational bottlenecks [13] with diminishing returns on performance [16]. Model reduction guided by controllability matrix rank was a topic of earlier work[17].

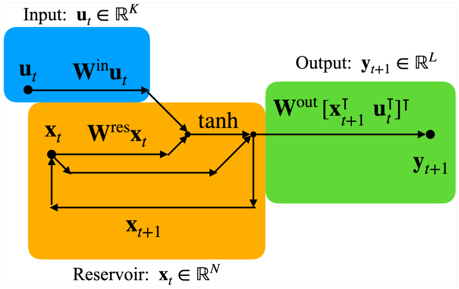
Input: $\mathbf{u}_t \in \mathbb{R}^K$
$\mathbf{u}_t$
$\mathbf{W}^{\text{in}}\mathbf{u}_t$
Output: $\mathbf{y}_{t+1} \in \mathbb{R}^L$
$\mathbf{W}^{\text{out}}\,[\mathbf{x}_{t+1}^\top,\ \mathbf{u}_t^\top]^\top$
$\mathbf{x}_t$
$\mathbf{W}^{\text{res}}\mathbf{x}_t$
tanh
$\mathbf{y}_{t+1}$
$\mathbf{x}_{t+1}$
Reservoir: $\mathbf{x}_t \in \mathbb{R}^N$

Fig. 1: Diagram of ESN

Here we illuminate some interesting connections between memory, controllability matrix rank, and the ESN operating near the edge of chaos.

2. ECHO STATE NETWORK DESIGN

ESNs operate between 3 spaces: input space $\mathcal{U} \subseteq \mathbb{R}^K$, reservoir state space $\mathcal{X} \subseteq \mathbb{R}^N$, and output space $\mathcal{Y} \subseteq \mathbb{R}^L$. The corresponding mappings between spaces are: input layer, recurrent reservoir, and trained output layer(see Fig. 1). These mappings are contained in the equations describing the ESN,

$$\mathbf{x}_{t+1} = F(\mathbf{x}_t, \mathbf{u}_t) = (1-\alpha)\mathbf{x}_t + \alpha\gamma \tanh(\mathbf{W}^{\mathbf{res}}\mathbf{x}_t + \mathbf{W}^{\text{in}}\mathbf{u}_t) \quad (1)$$

$$\mathbf{y}_{t+1} = \mathbf{W}^{\text{out}}\,[\mathbf{x}_{t+1}^\top\,\mathbf{u}_t^\top]^\top \quad (2)$$

Input mapping $\iota : \mathcal{U} \to \mathcal{X}$ scales and expands input from $\mathbb{R}^K$ into $\mathbb{R}^N$ via $\iota(\mathbf{u}_t) = \mathbf{W}^{\text{in}}\mathbf{u}_t$. Matrix $\mathbf{W}^{\text{in}} \in \mathbb{R}^{N\times K}$. Elements w_{ij}^{in} are drawn from uniform distribution $w_{ij}^{\text{in}} \sim \mathrm{U}(-r^{\text{in}}, r^{\text{in}})$ with parameter r^{in}. The transformed inputs in $\mathbb{R}^N$ perturb the current state $\mathbf{x}_t$ to generate $\mathbf{x}_{t+1}$.

Iterated mapping $\pi : \mathcal{X} \to \mathcal{X}$ given by $\mathbf{x}_t \mapsto \mathbf{x}_{t+1}$ is realized by a system of nodes acting as a reservoir of nonlinear dynamics. Node outputs are the interconnected elements of vector $\mathbf{x}_{t+1}$ resulting from (1). Elements of matrix $\mathbf{W}^{\mathbf{res}} \in \mathbb{R}^{N\times N}$ are random-distributed weights $w_{ij}^{\text{res}} \sim \mathrm{U}(-r^{\text{res}}, r^{\text{res}})$ connecting a complex network. Interconnection is accomplished through $\pi(\mathbf{x}_t) = \mathbf{W}^{\mathbf{res}}\mathbf{x}_t$.

The heart of an effective ESN is a random matrix $\mathbf{W}^{\mathbf{res}}$ called a reservoir. Creating effective ESN solves two tasks: (a) find an appropriate $\mathbf{W}^{\mathbf{res}}$ size $N \times N$ and (b) finding parameters inducing a topology on $\mathcal{X}$ appropriate for target data.

Both ι and π reside in the equation of motion (1). An element-wise application of tanh provides nonlinearity which is scaled against a "leak" term by α. A leak term admits direct influence from the previous state. Summing the terms creates state $\mathbf{x}_{t+1}$ on the manifold $\mathcal{X} \in \mathbb{R}^N$.

During operation the output layer (2) maps state update $\mathbf{x_{t+1}}$ and perturbation $\mathbf{u}_t$ to a prediction $\widehat{\mathbf{y}}_{\mathbf{t+1}}$ of the target $\mathbf{y}_{t+1} \in \mathcal{Y}$. If the reservoir is sufficiently complex then the output layer synchronizes the reservoir with the unknown system[18, 19, 20, 21, 22, 23].

3. TRAINING

Training the ESN relies on Tikhonov Regularization. The initialized ESN is given a random initial state $\mathbf{x}_0$ and $\mathbf{W}^{\text{out}}$. A pass is made through the training data to generate states $\mathbf{x}_t$. An initial number of $\mathbf{x}_t$ and all predictions $\widehat{\mathbf{y}_t}$ from the untrained $\mathbf{W}^{\text{out}}$ are ignored. The ignored $\mathbf{x}_t$ are a "burn-in" length T^{burn} where the ESN settles to states arising from "forcing" $\mathbf{u}_t$. These states are accumulated into matrix $\mathbf{\Phi}$.

Constructing $\mathbf{\Phi}$ involves stacking updates $\mathbf{x}_{t+1}$ onto $\mathbf{u}_t$ forming $\phi^{(t)} = [\mathbf{x}_{t+1}^\top\ \mathbf{u}_t^\top]^\top$. The ϕ's become the rows of matrix

$$\mathbf{\Phi} = \begin{bmatrix} —\phi^{(T^{\text{burn}})^\top}— \\ \vdots \\ —\phi^{(T^{\text{train}}-1)^\top}— \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{T^{\text{burn}}+1}^\top \ \mathbf{u}_{T^{\text{burn}}}^\top \\ \vdots \\ \mathbf{x}_{T^{\text{train}}}^\top \ \mathbf{u}_{T^{\text{train}}-1}^\top \end{bmatrix} \tag{3}$$

Where T^{train} is the training length. We then minimize the objective function using training data as target,

$$J(\mathbf{W}^{\text{out}}) = \|\mathbf{y} - \mathbf{\Phi}\mathbf{W}^{\text{out}\top}\|_2^2 + \varsigma\|\mathbf{W}^{\text{out}}\|_F^2. \tag{4}$$

The resulting output mapping $o : \mathcal{X} \oplus \mathcal{U} \to \mathcal{Y}$ given by $[\mathbf{x}_t^\top\ \ \mathbf{u}_t^\top]^\top \mapsto \mathbf{y}_{t+1}$ (equation (2)) synchronizes with training data. During prediction, $\widehat{\mathbf{y}}_{t+1}$ is recurrently fed back into the ESN as the next forcing $\mathbf{u}_t$.

4. THE ECHO STATE PROPERTY

A key characteristic of an ESN is- past states have diminishing effect on state updates and is called the Echo State Property:

Definition 1 (Echo State Property (ESP)[24]) An ESN whose reservoir dynamics are governed by $F(\mathbf{x}_t, \mathbf{u}_t)$ satisfies the ESP whenever for every initial conditions $\mathbf{x}_0, \mathbf{z}_0 \in X$, and for any input sequence of length T, i.e. $\mathbf{s}_T = (\mathbf{u}_t)_t^T$, it holds that $\|F(\mathbf{x}_0, \mathbf{s}_T) - F(\mathbf{z}_0, \mathbf{s}_T)\| \to 0$ as $T \to \infty$.

Reliably, an eigenvalue spectrum $\sigma(\mathbf{W^{res}})$ with spectral radius $\rho(\mathbf{W^{res}}) = \max(|\sigma(\mathbf{W^{res}})|) < 1$ has the ESP, however, diagonal Schur stability is sufficient [25]. For brevity $\rho(\mathbf{W^{res}}) = \rho$. This allows for ESP occurring when $\rho \geq 1$. Values of $\rho > 1$ are tested in our experiments.

To construct $\mathbf{W^{res}}$ we initialize a random matrix $\mathbf{W}$ and from this create a symmetric matrix,

$$\widetilde{\mathbf{W}} = (\mathbf{W} + \mathbf{W}^\top) - \mathbf{I}_N \operatorname{diag}(\mathbf{W}). \tag{5}$$

Then $\mathbf{W^{res}} = \frac{\eta}{\rho(\widetilde{\mathbf{W}})}\widetilde{\mathbf{W}}$ yields $\mathbf{W^{res}}$ with $\rho = \eta$.

5. CONTROLLABILITY MATRIX

From the linear system of control theory we extract the controllability matrix C_N and extend it to the nonlinear control system. A discrete-time linear system under forcing $\mathbf{u}_m$ is,

$$\mathbf{x}_{m+1} = \mathbf{A}\mathbf{x}_m + \mathbf{B}\mathbf{u}_m. \tag{6}$$

Iterating (6) over M steps yields a solution sequence or trajectory. For example, when $M = 3$ given $\mathbf{x}_0$ and forcings $(\mathbf{u}_m)_{m=0}^{M-1}$ is,

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{A}\mathbf{x}_0 + \mathbf{B}\mathbf{u}_0 \\ \mathbf{x}_2 &= \mathbf{A}\mathbf{x}_1 + \mathbf{B}\mathbf{u}_1 = \mathbf{A}^2\mathbf{x}_0 + \mathbf{A}\mathbf{B}\mathbf{u}_0 + \mathbf{B}\mathbf{u}_1 \\ \mathbf{x}_3 &= \mathbf{A}\mathbf{x}_2 + \mathbf{B}\mathbf{u}_2 = \mathbf{A}^3\mathbf{x}_0 + \mathbf{A}^2\mathbf{B}\mathbf{u}_0 + \mathbf{A}\mathbf{B}\mathbf{u}_1 + \mathbf{B}\mathbf{u}_2. \end{aligned}$$

Yielding a trajectory $(\mathbf{x}_i)_{i=1}^3$. For M-steps we have,

$$\mathbf{x}_M = \mathbf{A}^M\mathbf{x}_0 + \sum_{m=0}^{M-1} \mathbf{A}^{(M-m-1)}\mathbf{B}\mathbf{u}_m. \tag{7}$$

Representing the input sequence $(\mathbf{u}_i)_{i=0}^{M-1}$ as a vector $\mathbf{u} = [\mathbf{u}_0^\top, ..., \mathbf{u}_{M-1}^\top]^\top$ (7) becomes,

$$\mathbf{x}_M = \mathbf{A}^M\mathbf{x}_0 + [\mathbf{A}^{M-1}\mathbf{B} .... \mathbf{A}\mathbf{B}\,\mathbf{B}]\mathbf{u} \tag{8}$$

Notably in (7), columns of matrix $[\mathbf{A}^{M-1}\mathbf{B} .... \mathbf{A}\mathbf{B}\,\mathbf{B}]$ are formed by raising $\mathbf{A}$ to a power. We define the controllability matrix of (6) as[26],

Definition 2 (Controllability Matrix) Given input space $\mathbb{R}^K$, state space $\mathcal{X} \subseteq \mathbb{R}^N$, and matrices $\mathbf{A} \in \mathbb{R}^{N\times N}$ and $\mathbf{B} \in \mathbb{R}^{N\times K}$ the controllability matrix is defined,

$$C_N \triangleq [\mathbf{A}^{N-1}\mathbf{B} \ldots \mathbf{A}^2\mathbf{B}\ \mathbf{A}\mathbf{B}\ \mathbf{B}] \in \mathbb{R}^{N\times NK}. \tag{9}$$

Then (7) can be written,

$$\mathbf{x}_M = \mathbf{A}^M\mathbf{x}_0 + \mathbf{C}_M\mathbf{u}. \tag{10}$$

Matrix C_N identifies the set of reachable $\mathcal{R} \subseteq \mathcal{X}$ states of (6) arising from sequences $(\mathbf{u}_t)_{i=0}^M$ for all M. A few useful points are: (i) $\mathcal{R} = \operatorname{range}(C_N)$, (ii) C_N is a Krylov sub-space, and (iii) if $\operatorname{rank}(C_N) = M < N$ then only a subspace is reachable. Underlying (iii) is the Cayley-Hamilton theorem that suggests a minimum ESP or memory length corresponding to N independent columns. Ideally $M = N$. To illustrate, let λ_1 have $|\lambda_1| = \rho(\mathbf{W^{res}}) < 1$ then $|\lambda_1^k| \to 0$ as $k \to \infty$, where

$k \in \mathbb{N}^{<0}$. Another interpretation is that $\mathbf{x}_{m-k}$ has diminished effect or fading memory of $\mathbf{x}_{m+1}$.

For nonlinear systems a tangent approximation to (1) evaluated at $(\mathbf{x}_m, \mathbf{u}_m)$ is constructed from Jacobians $\nabla F(\mathbf{x}_m, \mathbf{u}_m)$ in the form of (6). Let $\mathbf{z}_m^{(i)} = \mathbf{W}^{\text{in}(i)}\mathbf{u}_m + \mathbf{W}^{\mathbf{res}\,(i)}\mathbf{x}_m$, for rows $i = 1, \ldots, N$. Then,

$$\widetilde{\mathbf{A}}(\mathbf{x}_m, \mathbf{u}_m) = \nabla_{\mathbf{x}_m}\mathbf{x}_{m+1} = (1-\alpha)\mathbf{I_N}$$
$$+\alpha\gamma \begin{bmatrix} w_{1,1}^{\text{res}(1)}\operatorname{sech}^2(\mathbf{z}_m^{(1)\top}) & \cdots & w_{1,N}^{\text{res}(1)\top}\operatorname{sech}^2(\mathbf{z}_m^{(1)\top}) \\ \vdots & \ddots & \vdots \\ w_{N,1}^{\text{res}(N)}\operatorname{sech}^2(\mathbf{z}_m^{(N)\top}) & \cdots & w_{N,N}^{\text{res}(N)}\operatorname{sech}^2(\mathbf{z}_m^{(N)\top}) \end{bmatrix} \tag{11}$$

and when K=1,

$$\widetilde{\mathbf{B}}(\mathbf{x}_m, \mathbf{u}_m) = \nabla_{\mathbf{u}_m}\mathbf{x}_{m+1} = \alpha\gamma \begin{bmatrix} w_1^{\text{in}}\operatorname{sech}^2(\mathbf{z}_m^{(1)\top}) \\ w_2^{\text{in}}\operatorname{sech}^2(\mathbf{z}_m^{(2)\top}) \\ \vdots \\ w_N^{\text{in}}\operatorname{sech}^2(\mathbf{z}_m^{(N)\top}) \end{bmatrix}. \tag{12}$$

Here, $\widetilde{\mathbf{A}}(\mathbf{x}_m, \mathbf{u}_m), \widetilde{\mathbf{B}}(\mathbf{x}_m, \mathbf{u}_m)$ (for brevity $\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}$) are functions of $(\mathbf{x}_m, \mathbf{u}_m)$. To calculate rank given an ESN, obtain

$$\widetilde{\mathbf{A}} = \nabla_{x_m}F\big|_{(\mathbf{x}_m,\mathbf{u}_m)}, \widetilde{\mathbf{B}} = \nabla_{u_m}F\big|_{(\mathbf{x}_m,\mathbf{u}_m)}. \tag{13}$$

Then at $(\mathbf{x}_m, \mathbf{u}_m)$ form,

$$\mathcal{C}_N = [\widetilde{\mathbf{A}}^{N-1}\widetilde{\mathbf{B}} \ \ldots \ \widetilde{\mathbf{A}}^2\widetilde{\mathbf{B}} \ \ \widetilde{\mathbf{A}}\widetilde{\mathbf{B}} \ \ \widetilde{\mathbf{B}}], \tag{14}$$

where N is the size of $\mathbf{W^{res}} \in \mathbb{R}^{N\times N}$. *Remark*: ESNs described by (1) with $\rho(\mathbf{W^{res}}) < 1$, have equilibrium point $(\overline{\mathbf{x}}, \overline{\mathbf{u}}) = (\mathbf{x}_m, \mathbf{u}_m) = (\mathbf{0}, \mathbf{0})$ such that,

$$\widetilde{\mathbf{A}}(\mathbf{0},\mathbf{0}) = \nabla_{x_m}F(\mathbf{0},\mathbf{0}) = (1-\alpha)\mathbf{I}_N + \alpha\gamma\mathbf{W^{res}} \tag{15}$$

$$\widetilde{\mathbf{B}}(\mathbf{0},\mathbf{0}) = \nabla_{u_m}F(\mathbf{0},\mathbf{0}) = \alpha\gamma\mathbf{W}^{\text{in}}. \tag{16}$$

At $(\mathbf{x}_m, \mathbf{u}_m) = (\mathbf{0}, \mathbf{0})$ there is no influence from location $\mathbf{x}_m$ or forcing $\mathbf{u}_m$ causing $\mathbf{z}_m^{(i)} = \mathbf{0}$. This leaves only the effects of parameters α, γ and properties of $\mathbf{W^{res}}$ and $\mathbf{W}^{\text{in}}$.

6. EXPERIMENTS

Lorenz-63 is generated from initial point $\mathbf{u}_0 = [1,1,1]^\top$ by *Scipy.integrate odeint*. Sequence length T^{test} is test data length.

- Lorenz-63 (signals denoted $Lv^{(1)}, Lv^{(3)}, L3\text{D}$):
  $T^{\text{burn}} = 100, T^{\text{train}} = 13900, T^{\text{test}} = 1000$
  $v_{t+1}^{(1)} = 10(v_t^{(2)} - v_t^{(1)})$
  $v_{t+1}^{(2)} = v_t^{(1)}(28 - v_t^{(3)}) - v_t^{(2)}$
  $v_{t+1}^{(3)} = v_t^{(1)}v_t^{(2)} - \frac{8}{3}v_t^{(3)}$

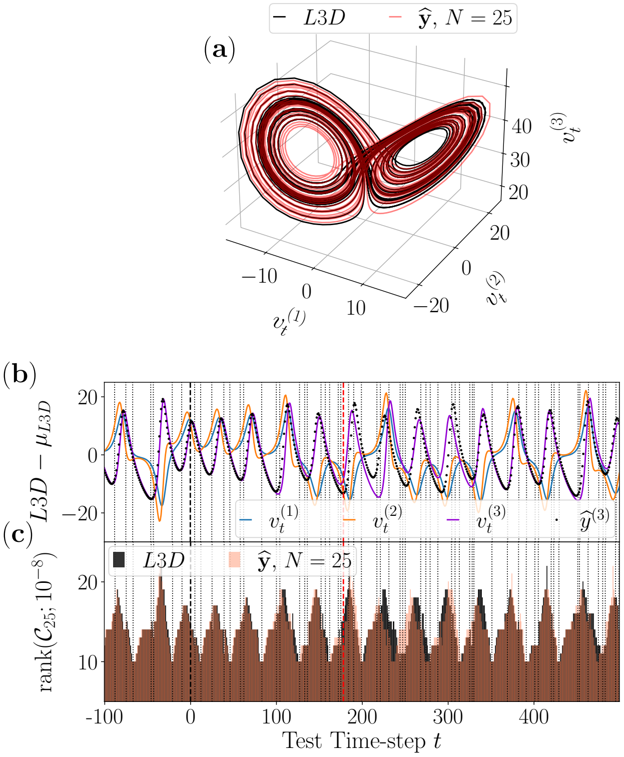

Fig. 2: (a) *L3D* (K=3 dimensions) target (black) and predictions (red) on test data plus free running ($T^{\text{test}} + 1000 = 2000$) from model size $N = 25$, $t \in [14000, 16000]$. (b)(c) show last 100 steps of training data and 500 steps $t \in [14000, 14500]$ from the $T^{\text{test}} = 1000$ sequence. (b) $Lv^{(3)} = \mathbf{y}$ (violet) and prediction $\widehat{\mathbf{y}}$ (black dotted) diverge at $t = 178$ (red dashed). Also shown are $v^{(1)}$ (blue) and $v^{(2)}$ (orange). (c) shows rank$(\mathcal{C}_{25}; 10^{-17})$ where $\mathcal{C}_{25} \in \mathbb{R}^{25\times 25}$, evaluated at each $(\overline{\mathbf{x}}_t, \mathbf{u}_t)$ for *L3D* and $(\overline{\mathbf{x}}_t, \widehat{\mathbf{y}}_t)$ (brown).

Normalized root mean squared error ($\mathcal{E}$) measures short-term error of test sequence $\mathbf{y}^{\text{test}}$ normalized by the standard deviation σ_{signal} over training and test sequences.

$$\mathcal{E}(\mathbf{y}^{\text{test}}, \widehat{\mathbf{y}}) = \frac{\sqrt{\frac{1}{T^{\text{test}}}\sum_{i=0}^{T^{\text{test}}-1}(y_i^{\text{test}} - \widehat{y}_i)^2}}{\sigma_{\text{signal}}}$$

To measure rank, $\mathcal{C}_N$ is normalized by its largest eigenvalue and *Numpy matrix_rank* is applied. We denote rank$(\mathcal{C}_N; \epsilon)$ if tolerance ϵ is set. The number of singular values greater than ϵ is rank$(\mathcal{C}_N; \epsilon)$. To take a series of snapshots we fix ϵ and solve for $\overline{\mathbf{x}}_t$ given $\mathbf{u}_t$. Evaluating $\mathcal{C}_N$ at points $((\overline{\mathbf{x}}_t, \mathbf{u}_t))_{t=T-k}^{T}$ where $k \in \mathbb{N}$, we create a series of rank$(\mathcal{C}_N; \epsilon)$ values.

Divergence between sequences is measured by cross-correlation between sequence values in $\mathbf{v}, \mathbf{w}$:

$$R_{\mathbf{vw}}[0] = \frac{\mathbf{v}^\top\mathbf{w}}{\|\mathbf{v}\|_2\|\mathbf{w}\|_2}. \tag{17}$$

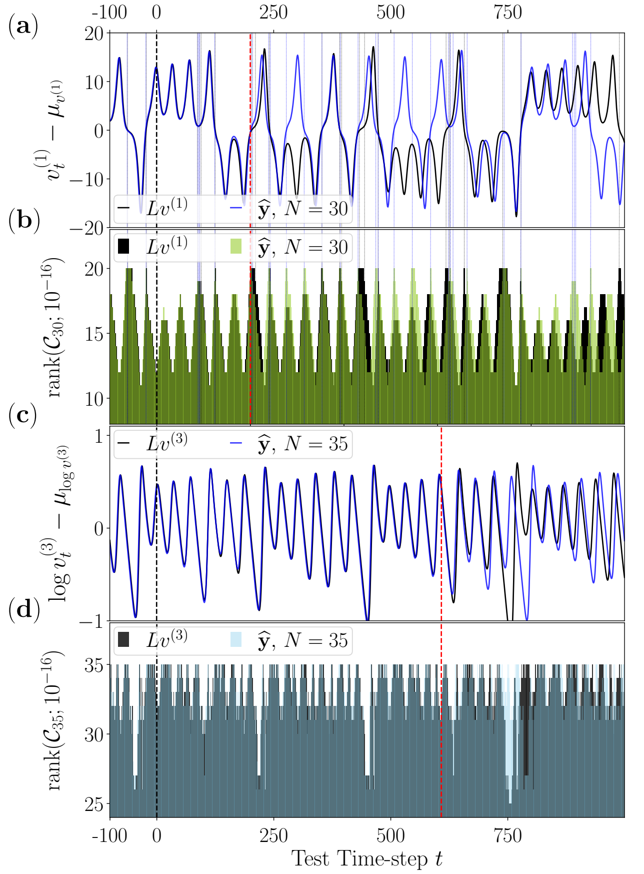

Fig. 3: Final 100 steps of training set followed by $T^{\text{test}} = 1000$ of test set $t \in [14000, 15000]$. Divergence threshold is correlation less than .95 (a) $Lv^{(1)} = \mathbf{y}$ (black) and $\hat{\mathbf{y}}$ (blue) diverge at t=200 (red dashed) for k=30. (c) $Lv^{(3)} = \log(v^{(3)}) - \mu_{v^{(3)}} = \mathbf{y}$ (black) and $\hat{\mathbf{y}}$ (blue) diverge at t=608 (red dashed) for k=35.

To calculate divergence we create vectors from sub-sequences $(v_i)_{i=t}^{t+l}$ and $(w_i)_{i=t}^{t+l}$ and $R_{\mathbf{vw}}[0]$ is stored for each t to create a sequence of correlations. From training data the minimum correlation value r_{vw} is found and a threshold is applied to predictions as a minimum correlation below which predictions have diverged. We use $l = N$ from $\mathbf{W^{res}} \in \mathbb{R}^{N \times N}$.

7. RESULTS

An equilibrium point is a dynamical system invariant satisfying $\bar{\mathbf{x}} = F(\bar{\mathbf{x}})$. These are either stable (attractive), denoted $\bar{\mathbf{x}}^s$, or unstable (repulsive) $\bar{\mathbf{x}}^u$. Trajectories of Lorenz-63 attractor states move between two lobes containing local $\bar{\mathbf{x}}^s$ separated by a single $\bar{\mathbf{x}}^u$ near the origin (after mean centering). In Fig. 2–3(c)(d) vertical dotted lines mark $(\bar{\mathbf{x}}_t, \mathbf{u}_t)$ (black) or $(\bar{\mathbf{x}}_t, \hat{\mathbf{y}}_t)$ (blue) where $\rho(\tilde{\mathbf{A}}) > 1$ indicating $\bar{\mathbf{x}}_t^u$ for the current t[27].

Figures 2(c),3(b)(d) show rank(C_N) varying along a series of snapshots. Clearly visible for $Lv^{(1)}$ in Fig. 3(a)(b), large rank, *inflection* points, and $\bar{\mathbf{x}}^u$ tend to coincide. Strangely, near $t = 750$ large rank and *local maximum* occur together. Looking deeper, lobe switches occur at $\mathbf{v}_t^{(1)} = 0 = \bar{\mathbf{x}}^u$, suggesting greater rank/memory encodes whether or not switching occurs. In contrast, at local extrema, low rank occurs. Old states are irrelevant to the quickly changing signal and so rank reduces. In counterpoint, reduced rank allows small divergence at peak or trough to propagate into full divergence.

Signal	N	$\rho(\mathbf{W^{res}})$	α	γ	$\rho(\tilde{\mathbf{A}}(\mathbf{0},\mathbf{0}))$	$\mathcal{E}$
$Lv^{(1)}$	30	1.31	.36	.9	1.05	.91
$Lv^{(3)}$	35	.88	.61	1.6	1.25	.69
$L3D$	25	.67	.68	3.8	2.03	.97

Table 1: Parameters $\rho(\mathbf{W^{res}})$ through γ are varied to find the best settings for each size N model tested. $\mathcal{E}$ is over T^{test}. Transformation of $Lv^{(3)}$ reversed for $\mathcal{E}$.

Figures 3(a)(b) for $t \in [780, 900]$ has $R_{\mathbf{u}\tilde{\mathbf{N}}}[0] = 0.91$ where $\tilde{\mathbf{N}}$ denotes a vector of rank values. Meaning rank matches increasing amplitude oscillations between $(u_t)_{t=780}^{900}$ and $(\tilde{N}_t)_{t=780}^{900}$. Also, divergence in Fig. 2(c),3(b)(d) means subsequent prediction rank will miss target signal rank.

In Table 1 we see only $Lv^{(1)}$ may not have the ESP due to $\rho(\mathbf{W}^{\text{res}}) > 1$. Looking at $\rho(\tilde{\mathbf{A}})$ without influence from position or forcing we see all snapshot systems are fundamentally unstable to different degree. Since Lorenz-63 is interrelated the single variable $Lv^{(3)}$ is under-observed. In Table 1, $\tilde{\mathbf{A}}(\mathbf{0},\mathbf{0})$ for both $Lv^{(3)}$ and $L3D$ are less stable than $Lv^{(1)}$. When predicting all three variables for $L3D$ inflection points will coincide with extrema, implying high and low memory needs coincide. Peaks in Fig.2(c) widen at these locations, e.g. $t = -50, 450$ in Fig. 2(b)(c).

These results suggest ESP/memory/rank(C_N) is not fixed at initialization but varies for dynamics needed. The rank(C_N) informs the range of ESP length varied through during prediction. Parameters play a key role in shaping the manifold of the reservoir and combine with the available rank so the ESN is unstable for appropriate $(\mathbf{x}_t, \mathbf{u}_t)$ and stable otherwise. From control theory we know that design of $\mathbf{W}^{\text{in}}$ and by extension $\mathbf{u}_t$ can make a rank deficient C_N full rank[28]. An effective ESN has a reservoir whose parameters shape manifold $\mathcal{X}$ so that rank varies with $\mathbf{u}_t$ appropriately.

8. CONCLUSION

We observe via rank(C_N) calculated at each forcing of test data that after ESN initialization ESP length is not static. The ESP varies depending on location in state space. Also, rank(C_N) indicates the length of ESP after which a forcing has diminished influence. Using rank(C_N) in this way illuminates the range of ESP length an effective ESN needs to predict a signal. As a benefit, burn-in length can be limited to the size of the reservoir matrix.

9. REFERENCES

[1] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *ICML*. 2013, vol. 28, p. III–1310–III–1318, JMLR.org.

[2] H. Jaeger, “The "echo state" approach to analysing and training recurrent neural networks-with an erratum note,” Tech. Rep. 148, GMD-Forschungszentrum Informationstechn, 2001.

[3] H. Jaeger, “Short term memory in echo state networks,” Tech. Rep. 152, GMD-Forschungszentrum Informationstechn, 2002.

[4] A. Ceni, P. Ashwin, and L. Livi, “Interpreting recurrent neural networks behaviour via excitable network attractors,” *Cogn. Comput.*, vol. 12, 03 2020.

[5] Nathaniel Rodriguez, Eduardo Izquierdo, and Yong-Yeol Ahn, “Optimal modularity and memory capacity of neural reservoirs,” *Netw. Neurosci.*, vol. 3, no. 2, pp. 551–566, 2019.

[6] P. Aceituno, G. Yan, and Y. Liu, “Tailoring echo state networks for optimal learning,” *iScience*, vol. 23, no. 9, pp. 101440, 2020.

[7] P. Barančok and I. Farkaš, “Memory capacity of input-driven echo state networks at the edge of chaos,” in *Artif. Neur. Netw. Mach. Learn., ICANN*, 2014.

[8] Goldmann M, F. Köster, K. Lüdge, and S. Yanchuk, “Echo state networks are universal,” *Chaos*, vol. 30, no. 9, pp. 093124, 2020.

[9] L. Gonon, L. Grigoryeva, and J. Ortega, “Memory and forecasting capacities of nonlinear recurrent networks,” *Physica D*, vol. 414, pp. 132721, 2020.

[10] N. Mayer, “Echo state condition at critical point,” *Entropy*, vol. 19, no. 1, pp. 3, 2017.

[11] T. Carroll and L. Pecora, “Network structure effects in reservoir computers,” *Chaos*, vol. 29, pp. 083130, 2019.

[12] A. Ferreira, T. Ludermir, and R. de Aquino, “An approach to reservoir computing design and training,” *Expert Syst. Appl.*, vol. 40, no. 10, pp. 4172–4182, 2013.

[13] A. Haluszczynski, J. Aumeier, J. Herteux, and C. Räth, “Reducing network size and improving prediction stability of reservoir computing.,” *Chaos*, vol. 30, no. 6, pp. 063136, 2020.

[14] A. Griffith, A. Pomerance, and D. Gauthier, “Forecasting chaotic systems with very low connectivity reservoir computers,” *Chaos*, vol. 29, no. 12, pp. 123108, 2019.

[15] J. Boedecker, O. Obst, J. Lizier, N. Mayer, and M. Asada, “Information processing in echo state networks at the edge of chaos,” *Theorie in den Biowissenschaften*, vol. 131, pp. 205–13, 2011.

[16] A. Chattopadhyay, P. Hassanzadeh, and D. Subramanian, “Data-driven prediction of a multi-scale lorenz 96 chaotic system using deep learning methods: Reservoir computing, ann, and rnn-lstm,” *Nonlin. Processes Geophys.*, vol. 27, no. 3, pp. 373–389, 2020.

[17] B. Whiteaker and P. Gerstoft, “Leaky integrator dynamical systems and reachable sets,” in *Proc. - ICASSP IEEE Int. Conf. Acoust. Speech Signal Process*, 2021, pp. 4025–4029.

[18] T. Weng, J. Song, H. Yang, C. Gu, J. Zhang, and M. Small, “Synchronization of reservoir computers with applications to communications,” *Physica A*, vol. 544, pp. 123453, 2020.

[19] Thomas Lymburn, D. Walker, M. Small, and T. Jüngling, “The reservoir’s perspective on generalized synchronization.,” *Chaos*, vol. 29 9, pp. 093133, 2019.

[20] A. Banerjee, J. Pathak, R. Roy, J. Restrepo, and E. Ott, “Using machine learning to assess short term causal dependence and infer network links,” *Chaos*, vol. 29, no. 12, pp. 121104, 2019.

[21] A. Cunillera, M. Soriano, and I. Fischer, “Cross-predicting the dynamics of an optically injected single-mode semiconductor laser using reservoir computing,” *Chaos*, vol. 29, no. 11, pp. 113113, 2019.

[22] J. Platt, A. Wong, R. Clark, S. Penny, and H. Abarbanel, “Robust forecasting through generalized synchronization in reservoir computing,” 2021.

[23] P. Antonik, M. Gulina, J. Pauwels, and S. Massar, “Using a reservoir computer to learn chaotic attractors, with applications to chaos synchronization and cryptography,” *Phys. Rev. E*, vol. 98, pp. 012215, Jul 2018.

[24] C. Gallicchio, “Chasing the echo state property,” in *27th Proc. Eur. Symp. Artif. Neural Netw., ESANN*, 2019.

[25] I. Yildiz, H. Jaeger, and S. Kiebel, “Re-visiting the echo state property,” *Neural networks*, vol. 35, pp. 1–9, 2012.

[26] E. Sontag, *Mathematical Control Theory Second Edition*, Springer, 1998.

[27] H. Sayama, *Introduction to the Modeling and Analysis of Complex Systems*, OpenSUNY Textbooks, 2015.

[28] S.L. Brunton and J.N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, Cambridge University Press, 2019.