Chaotic Systems and Neural Networks

Claudiu Craciun[1], Anirbit Mukherjee [2]

[1]Student in the Department of Computer Science, The University of Manchester, UK
[2]Department of Computer Science, The University of Manchester, UK
[*]Contact: claudiu.craciun@student.manchester.ac.uk, anirbit.mukherjee@manchester.ac.uk

1 Introduction

Chaotic dynamical systems have a strong dependence on initial conditions, making their future state hard to predict. They are believed to arise in scenarios such as weather forecasting.[Lor96] Numerical approaches for meteorological data prediction have started to be replaced by deep learning-based approaches.[SBG+21] In standard ML, the training data is sampled from the same data distribution the model is expected to predict on. However, in such "forecasting" tasks, the model is required to make predictions for future times outside the temporal support of the data distribution sampled from during training. Hence this is fundamentally different from standard ML. The deep-learning models studied here are tasked to approximate chaotic systems and thus can be seen as toy weather models. But, because of the analytic setup, we can do hard tests of performance like exploring the ability of the network to match known invariants of the dynamical system.

Two of the main measures of "chaos" are given by the Lyapunov Exponents and the Kaplan-Yorke dimension. These are some invariants that determine the stability and predictability of a system.[ABK91] Therefore, when training a neural network to learn dynamics beyond the training times, matching these values is a strong test of its performance.

Definition 1 (Lyapunov Spectrum)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}^d$ be a function with the associated Jacobian matrix Df. A d-dimensional discrete time dynamical system of the form $\boldsymbol{x}_{t+1} = \mathrm{f}^t(\boldsymbol{x}_0)$ with starting point $\boldsymbol{x}_0$ has d Lyapunov Exponents (LEs) and the i^{th} Lyapunov Exponent associated with $\boldsymbol{x}_0$ is given as,*

$$\lambda_i(\boldsymbol{x}_0) \coloneqq \lim_{t\to\infty} \frac{1}{2t} \ln\left(\mu_i(t, \boldsymbol{x}_0)\right)$$

where $\mu_i(t, \boldsymbol{x}_0)$ is the i^{th} largest eigenvalue of $Df^t(\boldsymbol{x}_0)^\top Df^t(\boldsymbol{x}_0)$. In the case of $d = 1$ this formula is equivalent to, $\lambda(f, x) = \lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} \log|f'(x_k)|$. A system is said to be chaotic if $\lambda_1 > 0$.

Definition 2. *The Kaplan-Yorke (KY) dimension of a multi-dimensional dynamical system is defined as,*

$$D_{KY} \coloneqq k + \sum_{i=1}^{k} \frac{\lambda_i}{|\lambda_{k+1}|}$$

where λ_i are the LEs and k is the largest integer so that the sum of the first k Lyapunov Exponents is positive.

In ergodic systems, the LEs do not depend on the choice of initial conditions,[VCC09] showing that both the LEs and the KY dimension are invariant measures of the system. All systems we will study here are ergodic.[SN79, MABB19, PV17, LM10]

Computation of the Lyapunov Exponents in Practice

- For estimating the LE of 1D systems such as the Logistic map $f(x) = 4x(1-x)$ and the Cubic map $f(x) = rx^3 + (1-r)x$, Definition 1 was used on a predicted/actual orbit x_i as, $\hat{\lambda}(f, x_0) = \frac{1}{T}\sum_{i=0}^{T} \ln|4 - 8x_i|$ and $\hat{\lambda}(f, x_0) = \frac{1}{T}\sum_{i=0}^{T} \ln\left|3rx_i^2 + 1 - r\right|$ respectively.
- For 2D systems, the Jacobian is $Df^t(\boldsymbol{x}_0) = Df(\boldsymbol{x}_{t-1})Df(\boldsymbol{x}_{t-2})\dots Df(\boldsymbol{x}_1)Df(\boldsymbol{x}_0)$. Direct usage of Definition 1 gave a good approximation for the first LE, but not the second one. The solution was to compute λ_1 and the sum $\lambda_1 + \lambda_2$:

$$\begin{aligned} \lambda_1 + \lambda_2 &= \frac{1}{2T}\ln|(\mu_1(t,\boldsymbol{x}_0))| + \frac{1}{2T}\ln|(\mu_2(t,\boldsymbol{x}_0))| = \frac{1}{2T}\ln|(\mu_1(t,\boldsymbol{x}_0)\mu_2(t,\boldsymbol{x}_0))| \\ &= \frac{1}{2T}\ln\left|\left(\det\left(Df^t(\boldsymbol{x}_0)^\top Df^t(\boldsymbol{x}_0)\right)\right)\right| = \frac{1}{T}\ln\left|\left(\det\left(Df^t(\boldsymbol{x}_0)\right)\right)\right| = \frac{1}{T}\sum_{i=0}^{T-1}\ln|\det(Df(\boldsymbol{x}_t))| \end{aligned} \tag{1}$$

- For the 3D Lorentz system, a discrete orbit $\boldsymbol{u}[i]$ of size $T = 400$ and separation $\Delta t = 0.01$ was generated. Three new orbits $\boldsymbol{q}_1[i], \boldsymbol{q}_2[i], \boldsymbol{q}_3[i]$ were created from some initial conditions such that $\{\boldsymbol{q}_1[0] - \boldsymbol{u}[0], \boldsymbol{q}_2[0] - \boldsymbol{u}[0], \boldsymbol{q}_3[0] - \boldsymbol{u}[0]\}$ forms an orthonormal set. Define $\boldsymbol{P}_1[i] = \boldsymbol{q}_1[i] - \boldsymbol{u}[i], \boldsymbol{P}_2[i] = \boldsymbol{q}_2[i] - \boldsymbol{u}[i], \boldsymbol{P}_3[i] = \boldsymbol{q}_3[i] - \boldsymbol{u}[i]$, three vectors that record the pointwise perturbations of $\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3$ from $\boldsymbol{u}$. After each 10 iterations, the perturbations were re-orthonormalized using the Gram-Schmidt process and their magnitudes were recorded in the arrays N_1, N_2, N_3. During the orthonormalisation step, let $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3$ be the last elements of $\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_3$. The update equations for the errors are,

$$
\begin{array}{lll}
1.\ n_1 \leftarrow \|\boldsymbol{p}_1\|_2 & 3.\ \hat{\boldsymbol{p}}_2{}' = \boldsymbol{p}_2 - (\boldsymbol{p}_2^T \hat{\boldsymbol{p}}_1)\hat{\boldsymbol{p}}_1 & 5.\ \hat{\boldsymbol{p}}_3{}' = \boldsymbol{p}_3 - (\boldsymbol{p}_3^T \hat{\boldsymbol{p}}_1)\hat{\boldsymbol{p}}_1 - (\boldsymbol{p}_3^T \hat{\boldsymbol{p}}_2)\hat{\boldsymbol{p}}_2 \\
2.\ \hat{\boldsymbol{p}}_1 \leftarrow \dfrac{\boldsymbol{p}_1}{n_1}, & 4.\ n_2 \leftarrow \|\hat{\boldsymbol{p}}_2{}'\|_2, \hat{\boldsymbol{p}}_2 \leftarrow \dfrac{\hat{\boldsymbol{p}}_2{}'}{n_2} & 6.\ n_3 \leftarrow \|\hat{\boldsymbol{p}}_3{}'\|_2, \hat{\boldsymbol{p}}_3 \leftarrow \dfrac{\hat{\boldsymbol{p}}_3{}'}{n_3}
\end{array} \tag{2}
$$

n_1, n_2, n_3 are recorded in N_1, N_2, N_3. The new start points for generating the next 10 values in the perturbed orbits are $\boldsymbol{q}_1[10i] = \boldsymbol{u}[10i] + \hat{\boldsymbol{p}}_1, \boldsymbol{q}_2[10i] = \boldsymbol{u}[10i] + \hat{\boldsymbol{p}}_2, \boldsymbol{q}_3[10i] = \boldsymbol{u}[10i] + \hat{\boldsymbol{p}}_3$. The LEs are computed using:[San96]

$$\lambda_i(f, \boldsymbol{u}[0]) = \frac{1}{T \times \Delta t} \sum_{k=1}^{T/10} \log(N_i[k]) \tag{3}$$

2 Experiments

A fully-connected neural network was used to approximate each of the first 4 systems in Table 1. Each net was a function $f : \mathbb{R}^d \to \mathbb{R}^d$, $f(\boldsymbol{x}) = \gamma(\boldsymbol{W}_3\sigma(\boldsymbol{W}_2\sigma(\boldsymbol{W}_1\boldsymbol{x})))$, where d is the dimension of the system, $\boldsymbol{W}_1 \in \mathbb{R}^{500\times d}, \boldsymbol{W}_2 \in \mathbb{R}^{500\times 500}$, $\boldsymbol{W}_3 \in \mathbb{R}^{d\times 500}$, $\sigma(x) = \frac{1}{1+e^{-x}}$ is the Sigmoid activation function applied component-wise and γ is the sigmoid function for the Logistic map and the linear activation for the Cubic, Henon and Standard maps. The loss function used to train the model was the Mean Squared Error Loss(MSE), $\text{loss}(\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1,\dots,T_{\text{train}}}) = \frac{1}{T_{\text{train}}} \sum_{i=1}^{T_{\text{train}}} \|f(\boldsymbol{x}_i) - \boldsymbol{y}_i\|_2^2$. An orbit of size $T = 100$ is generated by iterating f, starting from an initial point $\boldsymbol{x}_0$ different from the initial point used to generate the training data: $\hat{\boldsymbol{x}}_i = f(\hat{\boldsymbol{x}}_{i-1})$ for $0 < i \le T$. This predicted orbit is then plugged into the formulas from Section 1 to compute the LEs. Those algorithms are first validated by approximating the LE of the original system before being used for its neural surrogate.

Lorentz System setup: For this case we built a LSTM that predicts the next state of a discretized version of the system. An LSTM with a dense layer stacked at the end was used. The LSTM had a lookback window of 10 previous time steps and a hidden state $\boldsymbol{h}_k \in \mathbb{R}^{64}$, for $k = 0, \dots, 10$. Let $\text{LSTM}(\bar{\boldsymbol{x}}) \in \mathbb{R}^{64}$ be the output of the 10 LSTM cells,[HS97] where $\bar{\boldsymbol{x}}$ contains 10 consecutive positions. The k^{th} cell updates $\boldsymbol{h}_k$, and the output of the LSTM is $\boldsymbol{h}_{10}$. The fully connected layer at the end is used to bring the size 64×1 output back to 3×1. Let the last layer matrix be $\boldsymbol{W}_D \in \mathbb{R}^{3\times 64}$. The loss function is the MSE loss: $\text{loss}(\{\bar{\boldsymbol{x}}_i, \boldsymbol{y}_i\}_{i=1,\dots,T_{\text{train}}}) = \frac{1}{T_{\text{train}}} \sum_{i=1}^{T_{\text{train}}} ((\boldsymbol{W}_D \text{LSTM}(\bar{\boldsymbol{x}}_i)) - \boldsymbol{y}_i)^2$. The training data is 90% of an orbit of length $T_{\max} = 1000$ and separation $\Delta t = 0.01$. Four orbits are generated iteratively by the NN for computing the LEs as described in Eq. 3.

3 Results

For the systems in Table 1, the KY dimension of the neural approximation is similar to the real KY dimension, showing that they manage to capture the chaotic behavior from the systems they approximate.

Table 1: LEs and KY dimension of the neural approximations trained on 30% of the generated orbits (90% for Lorentz)

System	architecture	true KY	approx. KY(1)	KY(NN)(2)	true LE	approx. LE(1)	LE(NN)(2)
Logistic Map (1D)	1-500-500-1(3)	N/A	N/A	N/A	0.6931	0.6921	6527
Cubic Map (1D)	1-500-500-1	N/A	N/A	N/A	≈ 0.79[ZWZZ20]	0.77	0.76
Henon Map (2D)	2-500-500-2	1.26[KR20]	1.25	1.27	0.604,-2.34[KR20]	0.59,-2.33	0.65,-2.39
Standard Map (2D)	2-500-500-2	2[Spr04]	1.99	1.99	±0.10497[Spr04]	±0.039	± 0.095
Lorentz (3D)	LSTM	2.07[KR20]	2.06	2.01	2.16, 0, -32.4[KR20]	2.5,-0.4,-32.2	1.9,-1.8,-27.4

Note that some LEs are computed using $\ln$, while others using $\log_2$, depending on the source.
(1) This is an approximation based on real orbits from the system using the algorithms described in Section 1.
(2) This approximation uses the same algorithms, but is based on orbits generated by the neural network.
(3) A detailed explanation for the architecture can be found in Section 2.

References

ABK91 Henry DI Abarbanel, Reggie Brown, and MB Kennel. Lyapunov exponents in chaotic systems: their importance and their evaluation using observed data. *International Journal of Modern Physics B*, 5(09):1347–1375, 1991.

HS97 Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

KR20 Nikolay Kuznetsov and Volker Reitmann. *Attractor dimension estimates for dynamical systems: theory and computation*. Springer, 2020.

LM10 Zoran Levnajić and Igor Mezić. Ergodic theory and visualization. i. mesochronic plots for visualization of ergodic partition and invariant sets. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 20(3), 2010.

Lor96 Edward N Lorenz. Predictability: A problem partly solved. In *Proc. Seminar on predictability*, volume 1. Reading, 1996.

MABB19 Jeaneth Machicao, Marcela Alves, Murilo S Baptista, and Odemir M Bruno. Exploiting ergodicity of the logistic map using deep-zoom to improve security of chaos-based cryptosystems. *International Journal of Modern Physics C*, 30(05):1950033, 2019.

PV17 Ratna Pal and Kaushal Verma. Ergodic properties of families of h\'enon maps. *arXiv preprint arXiv:1708.06247*, 2017.

San96 Marco Sandri. Numerical calculation of lyapunov exponents. *Mathematica Journal*, 6(3):78–84, 1996.

SBG+21 Martin G Schultz, Clara Betancourt, Bing Gong, Felix Kleinert, Michael Langguth, Lukas Hubert Leufen, Amirpasha Mozaffari, and Scarlet Stadtler. Can deep learning beat numerical weather prediction? *Philosophical Transactions of the Royal Society A*, 379(2194):20200097, 2021.

SN79 Ippei Shimada and Tomomasa Nagashima. A numerical approach to ergodic problem of dissipative dynamical systems. *Progress of theoretical physics*, 61(6):1605–1616, 1979.

Spr04 Julien Clinton Sprott. Common chaotic systems. *Madison, WI*, 53706, 2004.

VCC09 Angelo Vulpiani, Fabio Cecconi, and Massimo Cencini. *Chaos: from simple models to complex systems*, volume 17. World Scientific, 2009.

ZWZZ20 Zhiqiang Zhang, Yong Wang, Leo Yu Zhang, and Hong Zhu. A novel chaotic map constructed by geometric operations and its application. *Nonlinear Dynamics*, 102:2843–2858, 2020.