Entropy and Information in Lorenz Trajectories

March 24, 2022

1 Lorenz '63 Model

The Lorenz model is one of the first simplified model used to describe atmospheric convection. Here we can find the system of equation that defines it.

$$\begin{cases} \dot{x} = \sigma(y - x) \\ \dot{y} = rx - y - xz \\ \dot{z} = xy - bz \end{cases} \tag{1}$$

Where σ is the Prandtl number, r is the Rayleigh number and $b > 0$ is a parameter linked to the ratio of the convective rolls. With $\sigma = 10$, $b = \frac{8}{3}$, $r = 28$ the system will show a chaotic behaviour.

Given a dynamical system $\dot{\boldsymbol{x}}(t) = \boldsymbol{F}\big(\boldsymbol{x}(t)\big)$, $\boldsymbol{x}^*$ is a fixed point if $\boldsymbol{F}(\boldsymbol{x}^*) = 0$. This system has three fixed points.

$$\begin{aligned} \boldsymbol{x}_0 =& (0, 0, 0) \\ \boldsymbol{x}_1 =& \big(\sqrt{b(r-1)}, \sqrt{b(r-1)}, r - 1\big) \\ \boldsymbol{x}_2 =& \big(\sqrt{b(r-1)}, \sqrt{b(r-1)}, r - 1\big) \end{aligned} \tag{2}$$

The origin, x_0, is a fixed point for each possible value of the parameters while $\boldsymbol{x}^*_{1,2}$ exist only if $r > 1$.

1.1 Linear Stability Analysis of fixed points

It is useful to study how the system behaves when it is close to one of its fixed point. We set

$$\boldsymbol{x}(t) = \boldsymbol{x}^* + \boldsymbol{\eta}(t)$$

and we assume that $\boldsymbol{\eta}(t)$ is small perturbation away $\boldsymbol{x}^*$. We substitute in $\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(\boldsymbol{x}(t))$ and by neglecting terms in order $\boldsymbol{\eta}^2$ we obtain the following equation, is the evolution in time of the small perturbation.

$$\dot{\boldsymbol{y}}(t) = \boldsymbol{A}\boldsymbol{y} \tag{3}$$

Where $A = \frac{\partial \boldsymbol{F}(\boldsymbol{x}^*)}{\partial \boldsymbol{x}_{ij}}$ with eigenvalues (λ_i). We say that $\boldsymbol{x}^*$ is linearly stable if $\Re(\lambda_i) \leq 0 \ \forall i$.

Stability of $\boldsymbol{x}_0^*$ We can see that if we remove the non linearity the dynamics of $z(t)$ is decoupled. So we can analyse the evolution of only $x(t)$ and $y(t)$:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} -\sigma & \sigma \\ r & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{4}$$

The trace of the matrix is $\tau = -\sigma - 1 < 0$, and determinant $\Delta = \sigma(1 - r)$. $\boldsymbol{x}_0^*$ is unstable if $r > 1$ ($\Delta < 0$ and $\tau < 0$ implies that $\lambda_i < 0 \ \forall i$).

Stability of $\boldsymbol{x}_{1,2}^*$ These points exists only if $r > 1$ and they are stable if:

$$1 < r < r_h = \frac{\sigma(\sigma + b + 3)}{\sigma - b - 1} \simeq 27.4 \tag{5}$$

where r_h is the value of r where the system shows its chaotic behaviour. In this case $r = 28$, this means that we are in chaotic regime.

1.2 Analysis of Phase Space

To produce the trajectories that has been used to study the phase space, the system has been solved numerically with RK4 method, with $\Delta t = 0.01$. We used these three following different type of trajectories.

- 9 random trajectories of 3000 points. This set is denoted with $\mathcal{D}_{rnd}$.
- 9 trajectories of 3000 points that begin close to the three fixed points. Namely 3 trajectories for each fixed point. This set is denoted with $\mathcal{D}_{FP}$.

In Fig.1 we see two sample of the two sets of trajectories.

As it has been shown in [2021], although the fixed and the random trajectories have the same amount of data, the performances of a Neural Network (LSTM) are better if the training is made with $\mathcal{D}_{FP}$ rather than the random ones.

1.3 SVD-Entropy

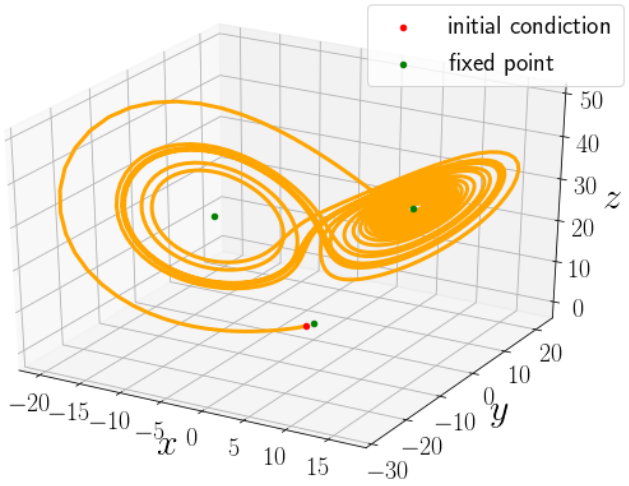
The amount of information that a trajectory contains, according to the Information Theory, is the minimum number of bit needed to reproduce it entirely.

To measure it we used the SVD-Entropy that it is based on Single Value Decomposition a method that has been widely used in different fields [1936],[2021]. We can define it as following:

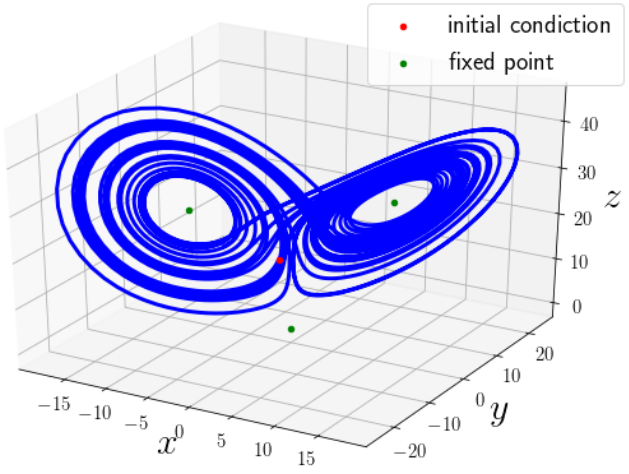
$$S_{svd} = S(\mathbf{x}(t), order, delay) \tag{6}$$

We denote with $\mathbf{x}$ the dataset that we want to study.

$$\begin{aligned} &\mathbf{x}(t_0) = x_0, \\ &\mathbf{x}(t) = (x_1, x_2, x_3, \ldots, x_N) \end{aligned} \tag{7}$$



(a)



(b)

Figure 1: (a) Trajectory sample of $\mathcal{D}_{FP}$ (b) Trajectory sample of $\mathcal{D}_{rnd}$

This function creates a matrix Y with the dataset.

$$\mathbf{y}(i) = (x_i, x_{i+\text{delay}}, ..., x_{i+(\text{order}-1)\text{delay}}) \tag{8}$$

$$Y = \begin{pmatrix} \mathbf{y}(1) \\ \mathbf{y}(2) \\ . \\ . \\ . \\ \mathbf{y}(N-(ord-1)del) \end{pmatrix}$$

If we use $order = 3$ and $delay = 1$ we obtain the following matrix:

$$Y = \begin{pmatrix} x_1 & x_2 & x_3 \\ x_2 & x_3 & x_4 \\ . & . & . \\ . & . & . \\ . & . & . \\ . & x_{N-1} & x_N \end{pmatrix}$$

The next step is to apply SVD decomposition to the matrix Y.

$$Y = U\Sigma V^* \tag{9}$$

Where U is an unitary matrix, V is also an unitary matrix, and Σ is a diagonal matrix. We denotes the eigenvalues of the matrix Σ with σ_i. After we compute the average eigenvalues:

$$\bar{\sigma}_k = \frac{\sigma_k}{\sum_j^M \sigma_j} \tag{10}$$

Where M is the number of eigenvalues.

After this we can compute the SVD Entropy:

$$S_{svd} = -\sum_k^M \bar{\sigma}_k log_2(\bar{\sigma}_k) \tag{11}$$

In figure 2 we have computed the average S_{svd} for 100 samples of 9 trajectories of $\mathcal{D}_{rnd}$ and $\mathcal{D}_{FP}$.

1.4 SVD-Entropy analysis of trajectories

We can study the $\mathcal{S}_{svd}$ of the trajectories to see how it grows increasing the number of points. In Fig 4 it has been shown the close to 200 number of points we can see that the amount of SVD-Entropy is bigger in the random trajectories. Fixed points trajectory shows to have less variance than the random one, except for the first 100 points, that shows higher fluctuations. To see what is the cause of higher fluctuations we can study the S_{svd} of trajectories close to $\boldsymbol{x}^*_{1,2,3}$ separately.

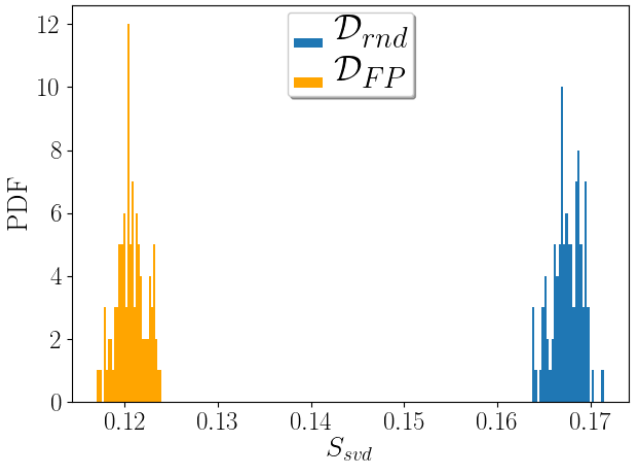


Figure 2: Histogram of SVD-Entropy of 100 trajectories for each type of dataset. It is shown that the fixed point dataset has less entropy

A we can see in fig (1.3) the fluctuation that has been noticed in Fig 4 are caused by the trajectories of the fixed point $\boldsymbol{x}_0^*$. As it is shown in Fig 6 the dynamics at the begin of $\boldsymbol{x}_0^*$ is different from $\boldsymbol{x}_{1,2}^*$. After almost 80 timesteps these dynamics are more similar. But at first $\boldsymbol{x}_0^*$ moves in a large space in the attractor, and faster. This is why we have this type of fluctuation. To be more confident with our results we can also repeat this anlysis using, instead of S_{svd} the Fisher Information, (see Appendix), and we obtain the same results.

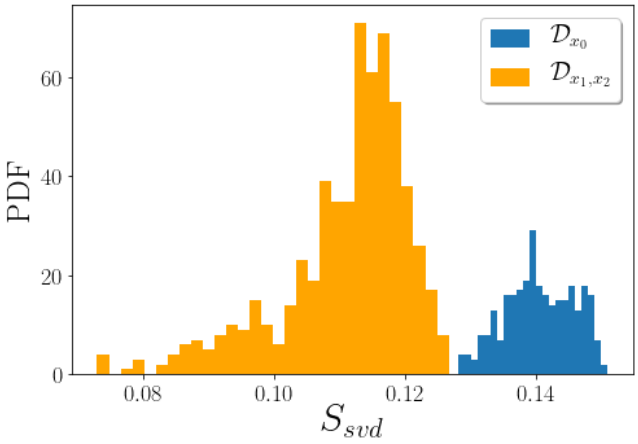


Figure 3: SVD-Entropy histogram of different fixed points trajectories. We can see that the $\mathcal{D}_{x_0}$ less informative

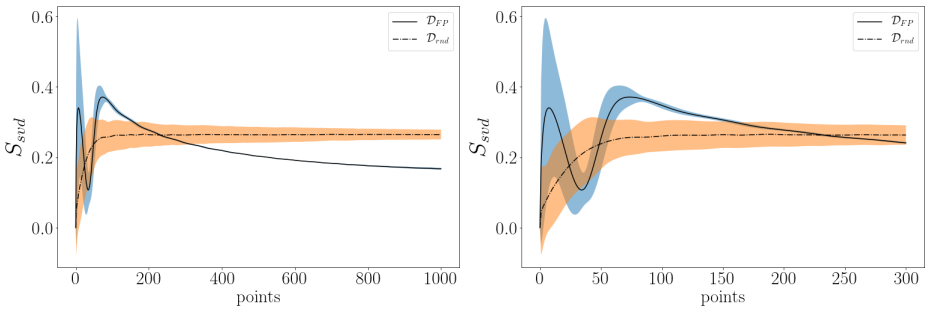
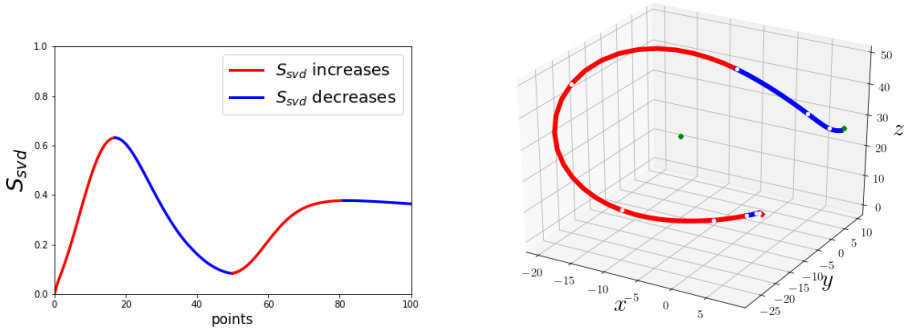
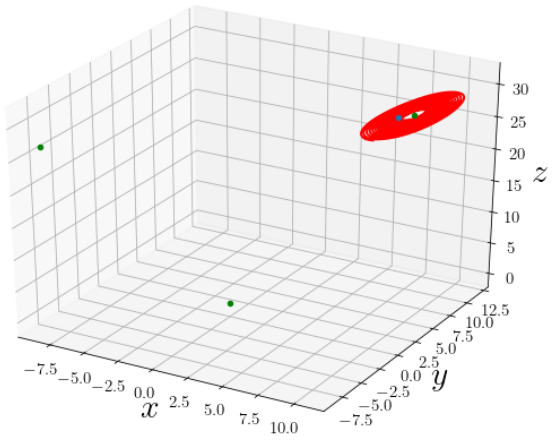


Figure 4: (a) Average SVD-Entropy of 100 trajectories in between of one standard deviation , (b) Zoom on the first points of SVD-Entropy

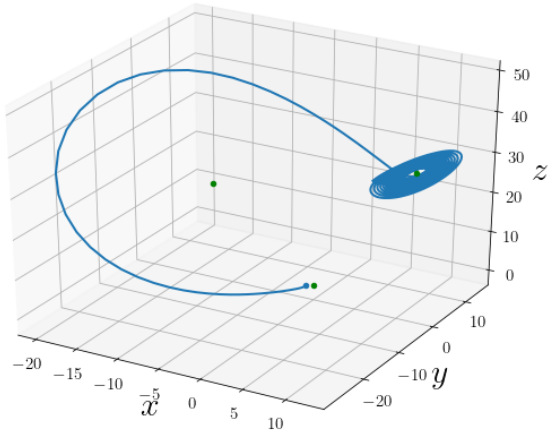


(a) (b)

Figure 5: (a) Evaluation of the avg SVD-Entropy in $\mathcal{D}_{x_{0,1,2}}$, (b) Trajectory from $\mathcal{D}_{x_0}$ the white points are plotted every 10 time steps



(a)



(b)

Figure 6: (a) Trajectory close to the fixed point $\boldsymbol{x}_1^*$ (b) Trajectory that begins close to zero

2 Machine Learning Approach

After we have showed that the set $\mathcal{D}_{FP}$ is the most informative, just analysing the trajectories, we can create a ML model and analyse it during the training process on the 2 different set of trajectories.

2.1 Models

We used two ML models to learn the Lorenz Model.

2.2 RNN

The first model is a standard RNN [Rumelhart et al. (1986)] with the following parameters:

- 1 layer is an RNN
- 2 layer is linear layer with in-features=50 out-features=3

2.3 MLP

The model that is used for training is a Multy Layer Perceptron [1958], with 3 layers:

- (layer 1): Linear(in-features=3, out-features=60)
- (layer 2): Linear(in-features=60, out-features=42)
- (layer 3): Linear(in-features=42, out-features=3)

2.4 Fisher Information Matrix

Given a dataset $\mathcal{D}$ we want to study the amount of information that a machine learning model extracts from it during the training process. To measure this quantity we use the Fisher Information Matrix.

Given a dataset $\mathcal{D} = \{x_i\}_{i=1}^N$ with $x \sim f(x|\theta)$, and $\hat{\theta}$ an estimator of θ: The theorem of Cramer-Rao says that:

$$\mathbb{E}[(\theta - \hat{\theta})^2] \geq \frac{1}{n\mathbb{E}\big[(\frac{\partial}{\partial\theta}\log f(x|\theta))^2\big]} \tag{12}$$

We can denote as the Fisher Information, the denominator of this equation. As the theorem says, the Fisher Information give an esteem of how far is the predicted parameter to the real one.

$$\mathcal{I}(\theta) := n\mathbb{E}_\theta\Big[\Big(\frac{\partial}{\partial\theta}\log f(x|\theta)\Big)^2\Big] \tag{13}$$

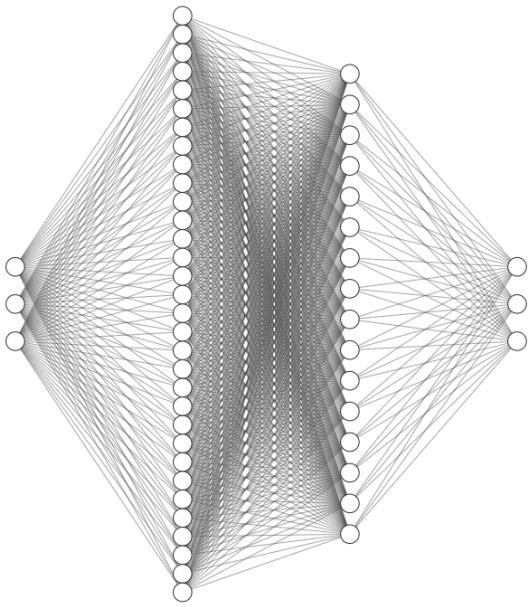


Figure 7: View of architecture of the MLP

If the distribution f is a function of more than one parameter, $\boldsymbol{\theta}$, we can write the Fisher Information Matrix as follows.

$$I(\boldsymbol{\theta}) := \mathbb{E}_{\boldsymbol{\theta}} \left[\left(\boldsymbol{\nabla}_{\boldsymbol{\theta}} \log f(\mathbf{x}; \boldsymbol{\theta}) \right)^2 \right] \tag{14}$$

The eigenvalues of this matrix show the most informative directions in parameter space. In practise we use the loss function as likelihood (MS2-error):

$$\mathcal{L}_i = \| \boldsymbol{x}_i - f\left(\mathbf{x}_i; \boldsymbol{\theta}\right) \|_2^2 \tag{15}$$

$$\widehat{f}\left(\widehat{\mathbf{x}}_i; \boldsymbol{\theta}\right) = \frac{\exp\left(-\mathcal{L}_i/\sigma^2\right)}{N^{-1} \sum_{j=1}^{N} \exp\left(-\mathcal{L}_j/\sigma^2\right)} \tag{16}$$

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}} \log f\left(\widehat{\mathbf{x}}_i; \boldsymbol{\theta}\right) \sim -\boldsymbol{\nabla}_{\boldsymbol{\theta}} \mathcal{L}_i + \frac{\sum_k \exp\left(-\mathcal{L}_k/\sigma^2\right) \boldsymbol{\nabla}_{\boldsymbol{\theta}} \mathcal{L}_k}{\sum_j \exp\left(-\mathcal{L}_j/\sigma^2\right)} \tag{17}$$

At the end the idea is to compute the (17) at the begin and after a certain number of epochs, to measure how the model extracts information from each dataset.

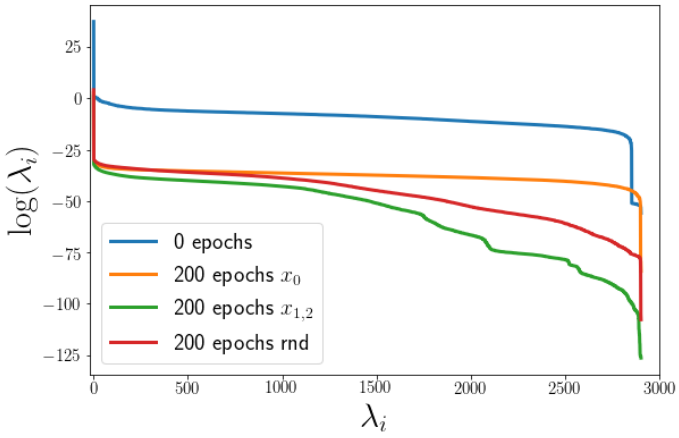


Figure 8: Fisher Information Matrix, spectrum of eigenvalues, for RNN trained with $\mathcal{D}_{x_0}, \mathcal{D}_{x_1}, \mathcal{D}_{x_2}, \mathcal{D}_{rnd}$. The model that extracts more information is the one who has been trained with $\mathcal{D}_{x_{1,2}}$

2.5 Results of Learning analysis

As we can see in fig(2.5) we can see that the eigenvalues are smaller in the model trained with $\mathcal{D}_{FP=x_0}$ than the one with other points. This means that the model is able to reach a flat surface during the training.

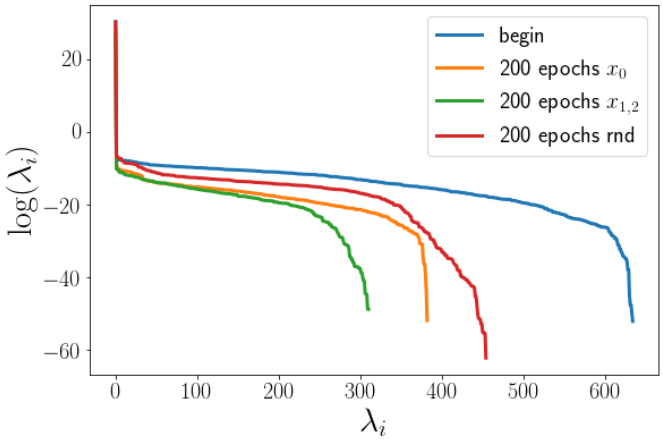


Figure 9: We can see that the model trained with $\mathcal{D}_{x_{1,2}}$ extracts more information than the others

3 Appendix

3.0.1 FISHER INFORMATION

3.1 Fisher Information

Fisher information is used together with svd decomposition.[**?**] We can define the Fisher Information in the following way[1]:

$$I_F = \sum_{k}^{M-1} \frac{(\bar{\sigma}_{k+1} - \bar{\sigma}_k)^2}{\bar{\sigma}_{k-1}} \tag{18}$$

The notation is the same that we have in SVD-Entropy. This represents the information that a trajectory contains, and its behaviour is the opposite of SVD-Entropy.

References

Bucci A., Semeraro O., Allauzen A., Chibbaro S., Mathelin L., 2021, arXiv preprint arXiv:2112.08458

Hotelling H., 1936, Biometrika, 28, 321

Raubitzek S., Neubauer T., 2021, Entropy, 23

Rosenblatt F., 1958, Psychological Review, pp 65–386

[1] for reference see here:`https://www.mdpi.com/1099-4300/23/11/1424`

Rumelhart D. E., Hinton G. E., Williams R. J., 1986, in Rumelhart D. E., Mcclelland J. L., eds, , Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations. MIT Press, Cambridge, MA, pp 318–362

Shannon C. E., 1948, The Bell System Technical Journal, 27, 379