Entropy and Information in Lorenz Trajectories

March 10, 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.

1.0.1 Fixed Points

Given a dynamical system

$$\begin{aligned} &\dot{\boldsymbol{x}}(t) = \boldsymbol{F}\big(\boldsymbol{x}(t)\big) \\ &\boldsymbol{x}^* \text{ is a fixed point if } \boldsymbol{F}(\boldsymbol{x}^*) = 0. \end{aligned} \tag{2}$$

This system has three fixed points.

$$\begin{aligned} \boldsymbol{x}_0 =& (0, 0, 0) \\ \boldsymbol{x}_1 =& \big(\sqrt{b(r-1)}, \sqrt{b(r-1)}, r - 1\big) \\ \boldsymbol{x}_2 =& \big(\sqrt{b(r-1)}, \sqrt{b(r-1)}, r - 1\big) \end{aligned} \tag{3}$$

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.

$$\dot{\boldsymbol{y}}(t) = \boldsymbol{A}\boldsymbol{y} \tag{4}$$

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

1.1.1 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 $y(t)$:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} -\sigma & \sigma \\ r & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{5}$$

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 $\lambda_i < 0 \forall i$).

1.1.2 Stability of $\boldsymbol{x}_{1,2}^*$

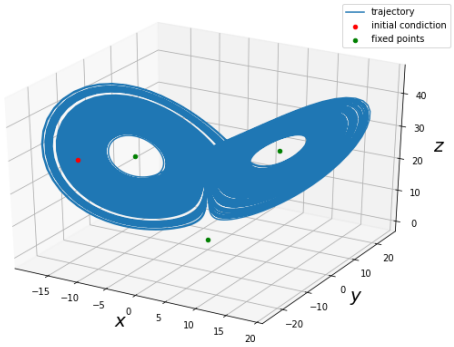
These points exists only if $r > 1$ and they are unstable:

$$\begin{aligned} r &= 28 \\ r_h &= \frac{\sigma(\sigma + b + 3)}{\sigma - b - 1} \simeq 27.4 \end{aligned} \tag{6}$$

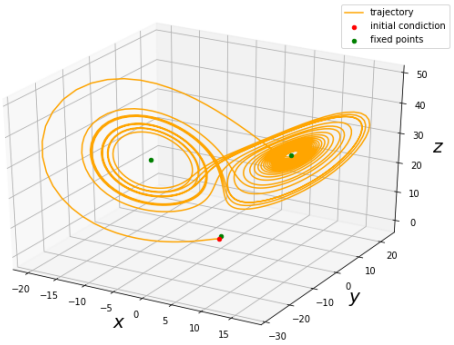
1.2 Dataset

To produce the dataset that has been used in the process of learning, the system has been solved numerically with RK4 method, with $\Delta t = 0.01$. The dataset is divided in two as follows.

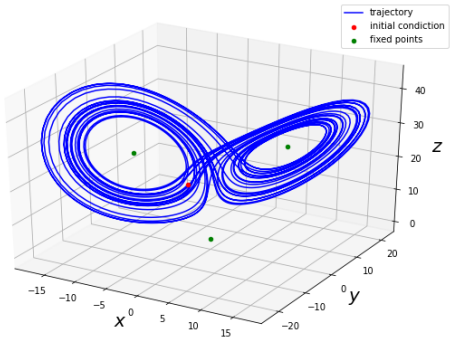
- $\mathcal{D}_{rnd} = \{\boldsymbol{x}(t_k)\}_{k=1}^N$ with N=3000 sampled by a 27000 points trajectory.
- $\mathcal{D}_{FP} = \{\boldsymbol{x}(t_k)\}_{k=1}^N$ N=3000, $\boldsymbol{x}(t_0)$ is close to one of the three fixed points $\boldsymbol{x}_0^*, \boldsymbol{x}_1^*, \boldsymbol{x}_2^*$.
- We can split in three $\mathcal{D}_{FP}$, namely one dataset for each fixed point : $\mathcal{D}_{x_0}$, $\mathcal{D}_{x_1}$, $\mathcal{D}_{x_2}$



(a)



(b)



(c)

Figure 1: (a) Long trajectory of 27,000 points, (b) Trajectory sample of $\mathcal{D}_{FP}$ (c) Trajectory sample of $\mathcal{D}_{rnd}$

2 Study the dataset

As it has been shown in (cita articolo), 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. This result could be explained because the system, close to fixed point, follows a dynamics far from the chaotic regime, because the non linear terms in these regions are neglected.

2.1 SVD-Entropy and trajectories

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 [1] that is inversely proportional to the Information. We can define it as following:

$$S_{svd} = S(\mathbf{x}(t), order, delay) \tag{7}$$

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

$$\begin{aligned} &\mathbf{x}(t_0) = x_0, \\ &\mathbf{x}(t) = (x_1, x_2, x_3, \dots, x_N) \end{aligned} \tag{8}$$

This function creates a matrix Y with the dataset.

$$\mathbf{y}(i) = (x_i, x_{i+\text{delay}}, ..., x_{i+(\text{order}-1)\text{delay}}) \tag{9}$$

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

If we use $order = 3$ and $delay = 1$ we obtain the following matrix[2]:

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

[1] Single Value Decomposition a method that has been widely used in different fields. In the context of Deep Learning it is used in Principal Component Analysis [1936] that is used in the field of dimensionality reduction. The key point behind this decomposition is to find the useful features that you need during the training process, namely the features that contain most of the signal you want to analyse. `https://raphaelvallat.com/entropy/build/html/generated/entropy.svd_entropy.html`

[2] To choose the delay and the order we can use the mutual information [1986] and to find the embedding dimension we can use the False Nearest Neighbours algorithm 2015, even if the qualitative behaviour of the results does not change even if we use different values of order and delay

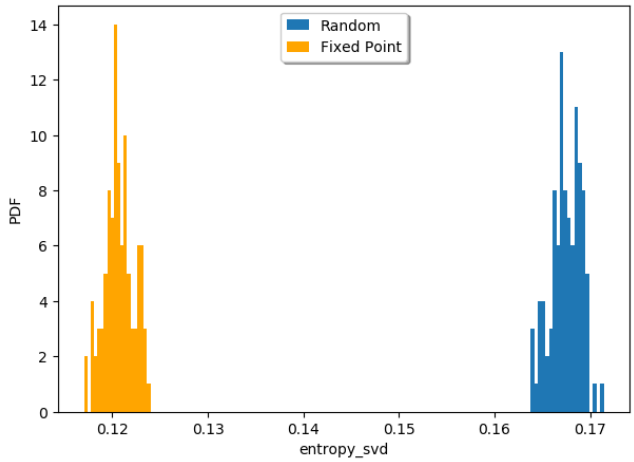


Figure 2: Histogram of SVD-Entropy of 100 trajectories for each type of dataset. It is shown that the fixed point dataset has less entropy

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

$$Y = U\Sigma V^* \tag{10}$$

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:

$$\bar{\sigma}_k = \frac{\sigma_k}{\sum_j^M \sigma_j} \tag{11}$$

Where M is the number of eigenvalues.

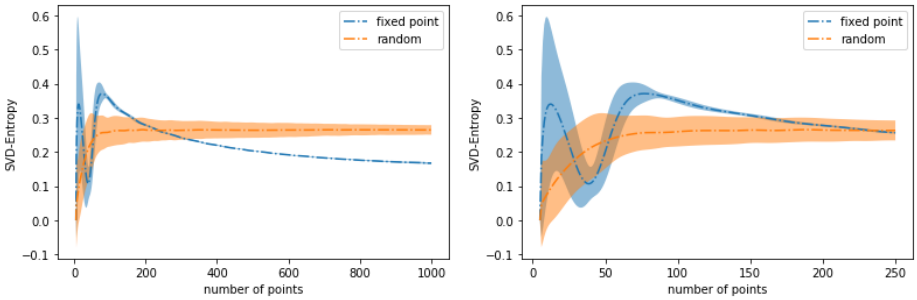
After this we can compute the SVD Entropy:

$$S_{svd} = -\sum_k^M \bar{\sigma}_k log_2(\bar{\sigma}_k) \tag{12}$$

2.2 SVD-Entropy as a function of datapoints

It is possible to follow a parallel approach if we study the SVD-Entropy as a function of the number of points of the trajectory.

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, in which the fluctuations are bigger. To understand these fluctuations we can study the trajectories close to $\boldsymbol{x}^*_{1,2,3}$ separately. A we can see in fig (2.2) the fluctuation that has been noticed in fig (3) are caused by the trajectories of the fixed point



(a) (b)

Figure 3: (a) Average SVD-Entropy of 100 trajectories in between of one standard deviation , (b) Zoom on the first points of SVD-Entropy

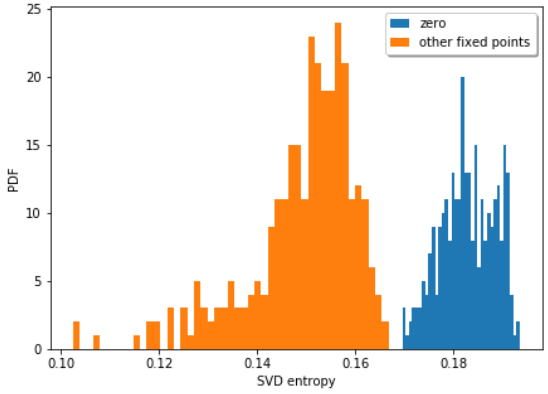
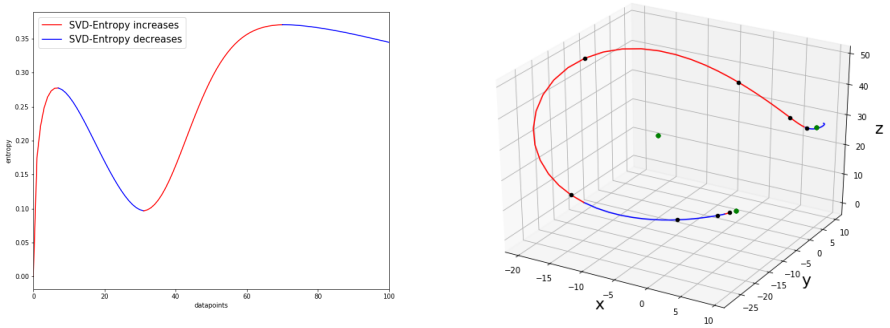


Figure 4: SVD-Entropy histogram of different fixed points trajectories. We can see that the $\mathcal{D}_{x_0}$ less informative

$\boldsymbol{x}_0^*$. As it is shown in fig (6) the trajectories from $\boldsymbol{x}_0^*$ are able to reach $\boldsymbol{x}_{1,2}^*$ only after almost 80 timesteps because before they move in a larger space in the attractor.



(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 black points are each 10 step

2.2.1 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.

2.2.2 Cramer-Rao Bound

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:

$$\mathbb{E}[(\theta - \hat{\theta})^2] \geq \frac{1}{n\mathbb{E}\big[(\frac{\partial}{\partial\theta}\log f(x|\theta))^2\big]} \tag{13}$$

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.

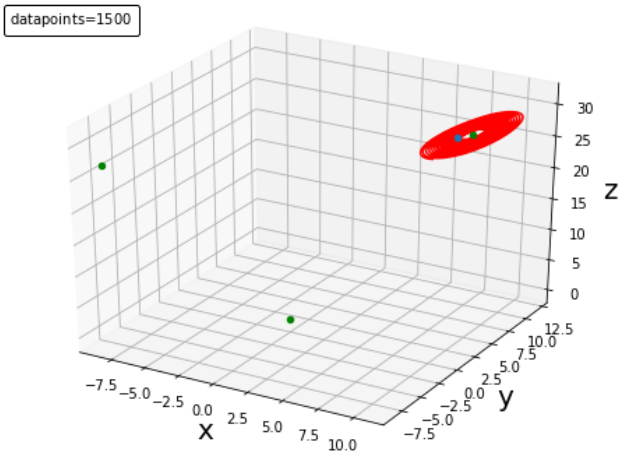
$$\mathcal{I}(\theta) := n\mathbb{E}_\theta\left[\left(\frac{\partial}{\partial\theta}\log f(x|\theta)\right)^2\right] \tag{14}$$

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

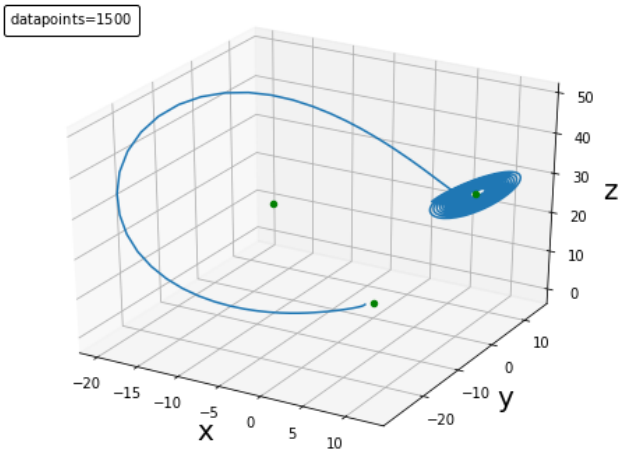
$$I(\boldsymbol{\theta}) := \mathbb{E}_{\boldsymbol{\theta}}\left[(\boldsymbol{\nabla}_{\boldsymbol{\theta}}\log f(\mathbf{x};\boldsymbol{\theta}))^2\right] \tag{15}$$

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

$$\mathcal{L}_i = \|\boldsymbol{x}_i - f(\mathbf{x}_i;\boldsymbol{\theta})\|_2^2 \tag{16}$$



(a)



(b)

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

$$\widehat{f}\left(\widehat{\mathbf{x}}_i;\boldsymbol{\theta}\right)=\frac{\exp\left(-\mathcal{L}_i/\sigma^2\right)}{N^{-1}\sum_{j=1}^{N}\exp\left(-\mathcal{L}_j/\sigma^2\right)} \tag{17}$$

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}}\log f\left(\widehat{\mathbf{x}}_i;\boldsymbol{\theta}\right)\sim-\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}_i+\frac{\sum_k\exp\left(-\mathcal{L}_k/\sigma^2\right)\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}_k}{\sum_j\exp\left(-\mathcal{L}_j/\sigma^2\right)} \tag{18}$$

At the end the idea is to compute the (18) at the begin and after a certain number of epochs, to measure how the model extracts information from each dataset.

2.3 Fisher Information Matrix

Then we can see how a RNN and an MLP extract information in different dataset type during the training in specific epochs during the training. Using the same amount of training points we can see different plots. It is better to be study the model when the loss function is low (under a certain threshold) because at the very begin of the training the model is not well formed.

2.4 RNN

The model is a standard RNN, with the first

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

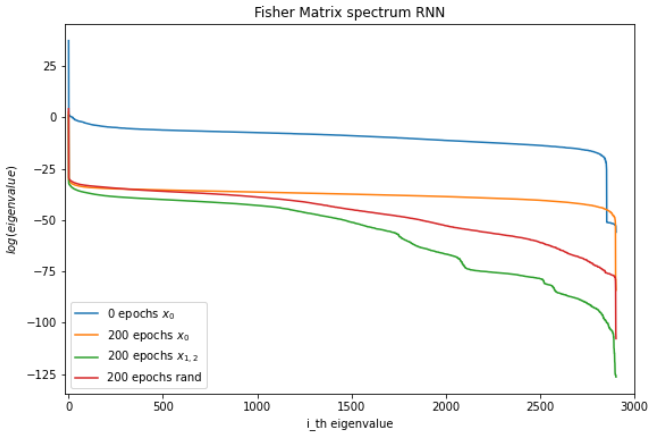


Figure 7: Fisher Information Matrix, spectrum of eigenvalues, for RNN trained with $\mathcal{D}_{x_0}, \mathcal{D}_{x_1}, \mathcal{D}_{x_2}, \mathcal{D}_{rnd}$

As we can see in fig(2.4) we can see that the eigenvalues are smaller in the model trained with $\mathcal{D}_{FP=x_0}$ than the one with other points.

2.5 MLP

The model that is used for training is a MLP, 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)

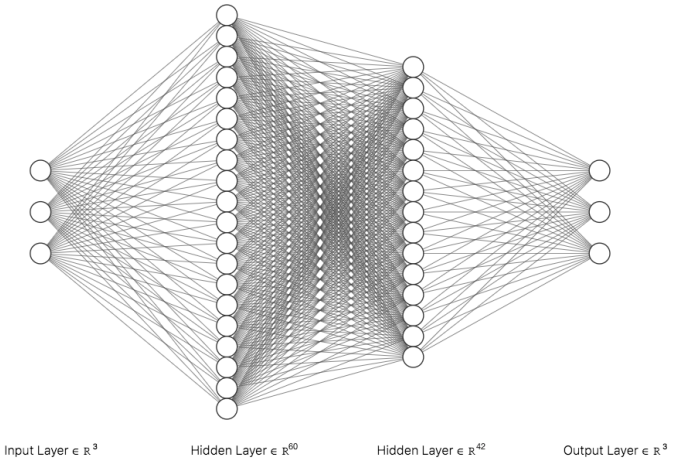


Figure 8: View of architecture of MLP

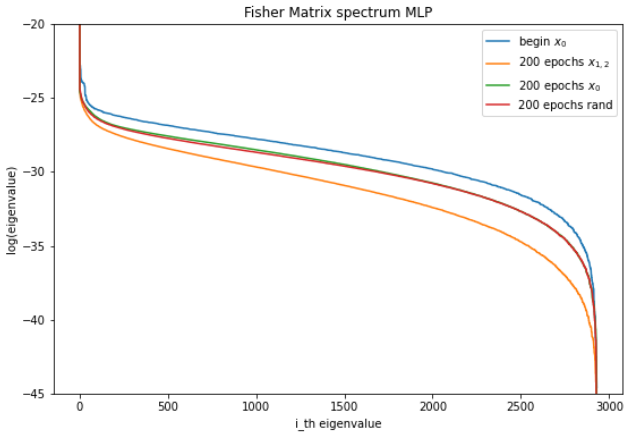


Figure 9: We can see that the model trained with $\mathcal{D}_{x_0}$ extracts more information than the others

3 Appendix

3.0.1 FISHER INFORMATION

3.1 Fisher Information

Fisher information is used together with svd decomposition.[2006] We can define the Fisher Information in the following way[3]:

$$I_F = \sum_{k}^{M-1} \frac{(\bar{\sigma}_{k+1} - \bar{\sigma}_k)^2}{\bar{\sigma}_{k-1}} \tag{19}$$

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

Fraser A. M., Swinney H. L., 1986, Phys. Rev. A, 33, 1134

Hotelling H., 1936, Biometrika, 28, 321

Mayer A. L., Pawlowski C. W., Cabezas H., 2006, Ecological Modelling, 195, 72

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

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

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

Tang Y., Krakovská A., Mezeiová K., Budáčová H., 2015, Journal of Complex Systems, 2015, 932750