Entropy and Information in Lorenz Trajectories

March 2, 2022

1 Lorenz '63 Model

The model we are going to study is the Lorenz model, one of the first simplified model used to describe atmospheric convection. Here we can find the system of equation that describes this model.

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

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 $\sigma = 10$, $b = \frac{8}{3}$, $r = 28$ it is known that the system will show a chaotic behaviour.

1.0.1 Fixed Points

Given a dynamical system $\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(\boldsymbol{x}(t))$, $\boldsymbol{x}^*$ 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. $x_{1,2}$ 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 $A = J(\boldsymbol{F}(\boldsymbol{x}^*))$ 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 this system:

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

The trace of the matrix is $\tau = -\sigma - 1 < 0$, and determinant $\Delta = \sigma(1 - r)$. If $r > 1$ $\boldsymbol{x}_0$ is unstable ($\Delta < 0$ and $\tau < 0$).

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

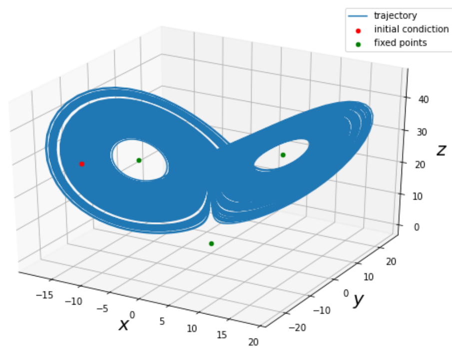
These points exists only if $r > 1$ and they are unstable:

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

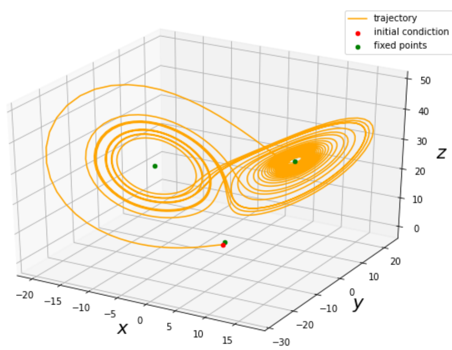
2.1 Dataset

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 produced trajectories are the following:

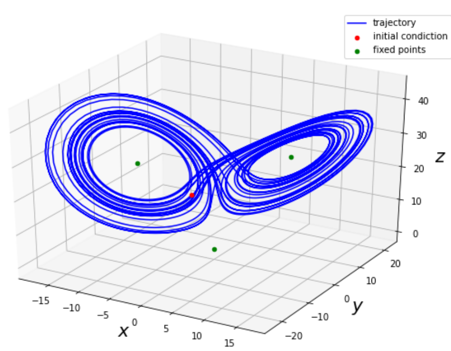
- 9 trajectories of 3000 points, sampled by a 27000 points trajectory.
- trajectories that originates from regions close to fixed points $\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2$



(a)



(b)



(c)

Figure 1: (a) Long trajectory of 27,000 points, (b) trajectory of 3000 points that starts close to a fixed point (c) Sampled trajectory of 3000 points

3 Study the dataset

As it has been shown in (cita articolo), although the fixed and the random trajectories have the same dimension, the performance of a Neural Network (LSTM) are better if the training has been made by using the fixed points trajectories rather than the random trajectories. The aim of this paper is also to understand why this happens. This results could be explained because the system close to fixed point follows a dynamics that is far from the chaotic regime. Namely close to fixed point the nonlinear terms are neglected.

3.1 SVD-Entropy and trajectories

To understand the differences in perfomances it has been studied the information that the different trajectories contain. The amount of information that a list of number contains is linked to its entropy. To measure it we used the SVD-Entropy [1]. That we can define as following:

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

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

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

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[2]:

[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 explain the variance 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

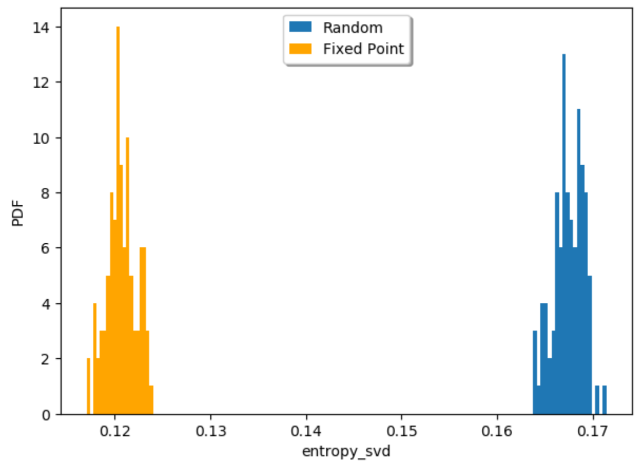


Figure 2: Histogram of SVD-Entropy of 100 trajectories for each type of dataset. It is shown that the fixed point dataset has less entropy, it means that it is contains more information.

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

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. Now we denotes the eigenvalues of the matrix Σ with σ_i. Now we can compute the average eigenvalues:

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

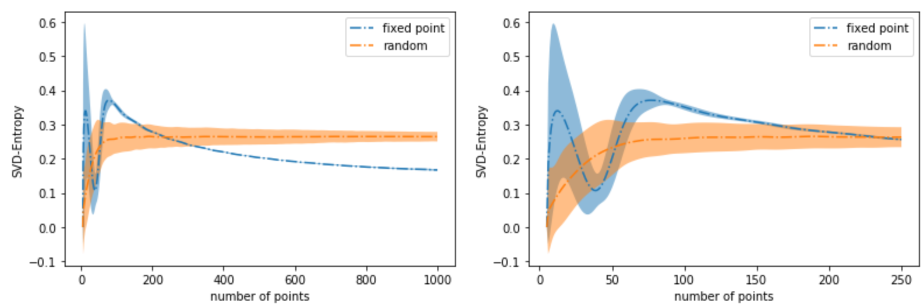
Where M is the number of eigenvalues.

After this we can compute the SVD Entropy:

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

3.2 Growth of Information and Entropy

We can try to see the growth of information when we increase the number of trajectory's points.
If we look at this by a statistical point of view, it could be useful to understand



(a) (b)

Figure 3: (a) Average of SVD-Entropy with one std , (b) Zoom on the first points of SVD-Entropy

the average behaviour of the IF and SVD entropy, and also to understand the fluctuation that we have. We can try to compare the different shapes of the average amount of information (or entropy) of different trajectories when we increase the number of points.

It is interesting to notice that around 200 number of points we can see that the amount of FI is bigger in the manifold data than in the part. We have the same behaviour with the svd entropy, but as we expect it is in the opposite direction.

It is important also to study the fluctuation of information that we have. We can also notice by looking at these plots that the part trajectories seems to have a more stable variation of information than the manifold. In particular we see that at the very beginning of the curve, namely around 200 points the svd entropy of the manifold dataset increases a lot. ——- THREE FIXED POINTS ———

3.3 WHY THIS FLUCTUATION ?

We can try to measure the amount of information that we obtain if we use a trajectory close to zero to the train our model. By following the same ideas that helped us in the previous section we are going to see the results of this computation. It has been used the same amount of data to obtain these two plots.

Here we can see also two different trajectories in the attractor.

These two trajectories have different svd entropy growth.

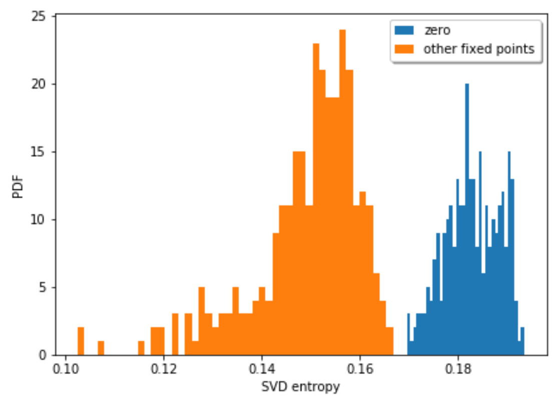
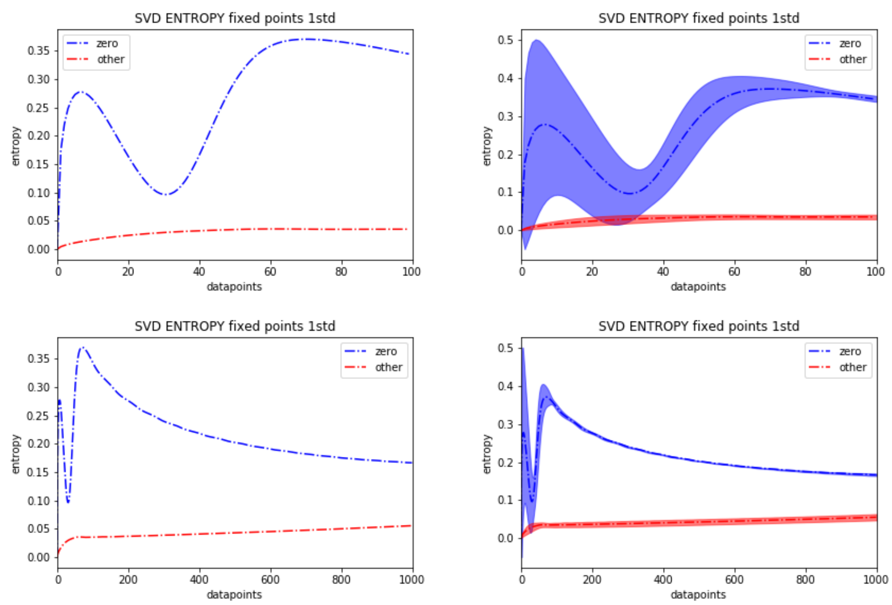
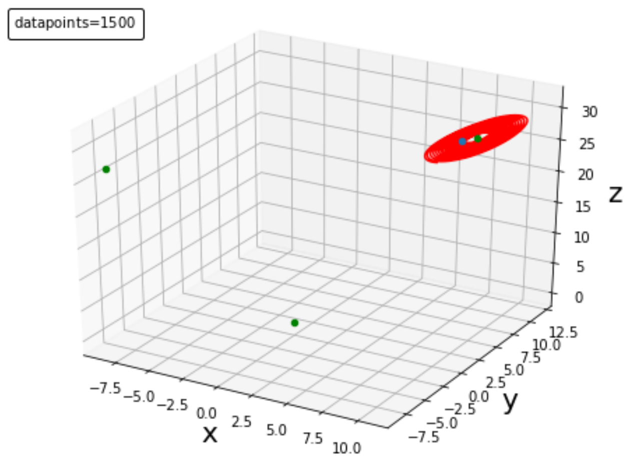
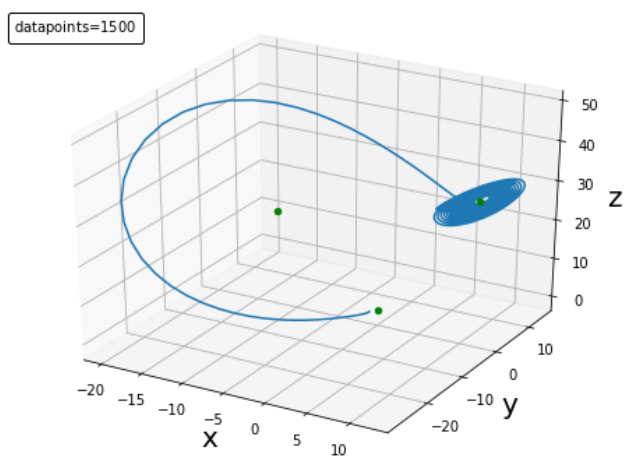


Figure 4: SVD-Entropy histogram of different fixed points trajectories.





(a)



(b)

Figure 5: (a) Trajectory close to the fixed point $\boldsymbol{x}_1$ (b) Trajectory that begins close to zero

3.3.1 FISHER INFORMATION MATRIX

To study the differences between two type of trajectories we can analyze the difference of the amount of information that the model extracts during the training process.

3.3.2 Theory of Information Matrix

We have a Dataset: $\mathcal{T} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{N}$ and we want to study the amount of information that a model (for instance LSTM) extracts during the training process from these data. We denote with $f(\hat{\mathbf{y}};\theta)$ the density function of a random variable $\hat{\mathbf{Y}}$ parametrized by θ. We are interested in the shape of likelihood function. That is why we need the Fisher Matrix, that it is defined in this expression:

$$I(\boldsymbol{\theta}) := \mathbb{E}_{\widehat{Y}}\left[\left(\boldsymbol{\nabla}_{\boldsymbol{\theta}} \log f(\widehat{\mathbf{y}};\boldsymbol{\theta})\right)^2\right] \tag{12}$$

This matrix, with its eigenvalues is able to describe the 'direction' that contains more information in the space of parameters. In practice, in a ML approach, we want to know the gradient of the loss function, $\mathcal{L}(\theta)$, with respect to model parameters.

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}} \log \widehat{f}\,(\widehat{\mathbf{y}}_i;\boldsymbol{\theta}) \sim -\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}_i + \frac{\sum_k \exp\left(-\mathcal{L}_k/\sigma^2\right) \boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}_k}{\sum_j \exp\left(-\mathcal{L}_j/\sigma^2\right)} \tag{13}$$

3.3.3 Cramer-Rao Bound

3.3.4 Fisher Matrix Plots

Then we can see how a RNN extracts information in different dataset type after 10 epochs. Using the same amount of training points we can see different plots.

——INTRODURRE DISCORSO DELLA LOSS FUNCTION ————

We can see that in this plot the amount of information extracted from the dataset "mani" , namely trajectories closes to fixed points, is greater than then the others dataset.

Here we have a zoom of the spectrum of the eigenvalues of the FIM of different trajectories after 10 epochs. As we can see the dataset that contains the bigger amount of information is "mani". Indeed the eigenvalues of the FIM are linked with the shape of the space in which the "gradient descent" moves.

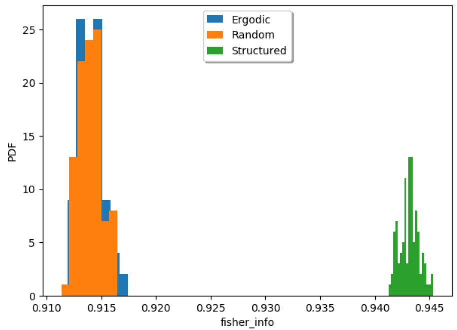
3.3.5 FIM and information of zero

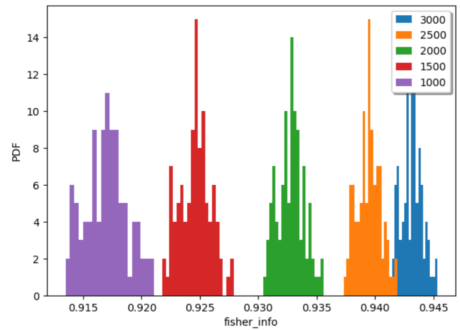
To answer to the question of this section we can use the FIM approach to see how this matrix changes if we use as training set the trajectories close to zero, and the others. We can find these results in the following plots.

As we can see the dataset of trajectories closes to zero seems to contain less information than the others. Indeed we see that after 10 epochs the green curve is close to zero.

4 Appendix

4.0.1 FISHER INFORMATION





4.1 Fisher Information

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

$$I_F = \sum_{k}^{M-1} \frac{(\bar{\sigma}_{k+1} - \bar{\sigma}_k)^2}{\bar{\sigma}_{k-1}} \tag{14}$$

The quantities are exactly the same that we have defined in the previous section. The difference between this function and the svd entropy is that FI behaves contrary to the entropy. We can see this in the plots.[2021]

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

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

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