SAPIENZA
UNIVERSITÀ DI ROMA

Exploring regime shift through chaos-driven neural network

Facoltà di Scienze Matematiche, Fisiche e Naturali
Fisica

Luca Boerio
ID number 1845870

Advisor
Vittorio Loreto

Co-Advisor
Alessandro Londei

Academic Year 2023/2024

Thesis defended on 21/10/2024
in front of a Board of Examiners composed by:

Prof. (chairman)
Prof.

Exploring regime shift through chaos-driven neural network
Master thesis. Sapienza University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Version: September 30, 2024

Author's email: boerio.1845870@stundenti.uniroma1.it

Abstract

This thesis explores regime shift phenomena and their prediction using chaos-driven neural networks. Regime shifts refer to abrupt and significant changes in the structure of complex systems such as ecosystems, climate, or financial systems, marked by transitions between order and chaos. By employing neural networks regularized with Lyapunov exponents, the study demonstrates how chaos can enhance the network's exploratory capabilities, enabling better handling of sudden transitions and adaptation to dynamic changes. The results show significant performance improvements, especially in chaotic systems, experimentally validating the effectiveness of the proposed method. The potential applications of this technique span across various fields, including continual learning and innovation challenges.

Contents

Chapter 1

Introduction

The main objects of this work are the wide category of regime shift phenomena and the forecasting of these phenomena. Regime shifts are large, abrupt changes in the structure or trend of ecosystems, climate, financial systems, or complex, physical, dynamical systems; in general, it regards all those situations in which a system is at a critical point, where even the smallest perturbation can dramatically change the course of that phenomenon, as shown in Figure 1.2([10]). This topic has crucial importance due to its presence in our everyday life. Indeed, there has been abundant suggestive evidence that many natural systems can be found at a critical point between order and disorder [11]: we can find this behavior in brain activity, genetics, cardiac arrhythmia, cells' collective behavior, and collective motion, like that of bird flocks, insect swarms, or mammal herds. Moreover, it seems that criticality is not something rare, but instead is the favored mechanism through which life evolves [14]: with this self-organized criticality, biological systems maximize their adaptability and survival, regulating their parameters in order to exhibit a critical, ordered or disordered behavior, depending on the situation.
Usually one think criticality and phase transition in statistical terms, like the mean behavior of certain quantities or correlation functions, but we can think of it also in a dynamical way.

Indeed, systems not only can undergo order-disorder transitions but also order-chaos transitions, and more specifically chaotic systems have a sensitive dependence on initial conditions, meaning that a minimal difference in the dynamics' initial condition will bring the system exponentially far from the reference trajectory. Like one can see in Figure 1.1 , examples of these systems are climate, turbulent fluids, and neurons dynamics.
These sudden, unpredictable changes during the evolution of a certain phenomenon may remind of one of the most important topics in complex systems, which is novelties and innovation. As it is said in [12], "they can be viewed as first-time occurrences of something at the individual or collective level" just like the sudden changing of paradigms in some phenomena, or like systems that exhibit a completely new transition phase. It would be a great step forward to build something able to elastically adapt to these novelties, these changes; such a machine would have to figure out, on the basis of actual data, the possibilities that may arise from this actual world. This concept is well captured by the notion of Adjacent Possible, first

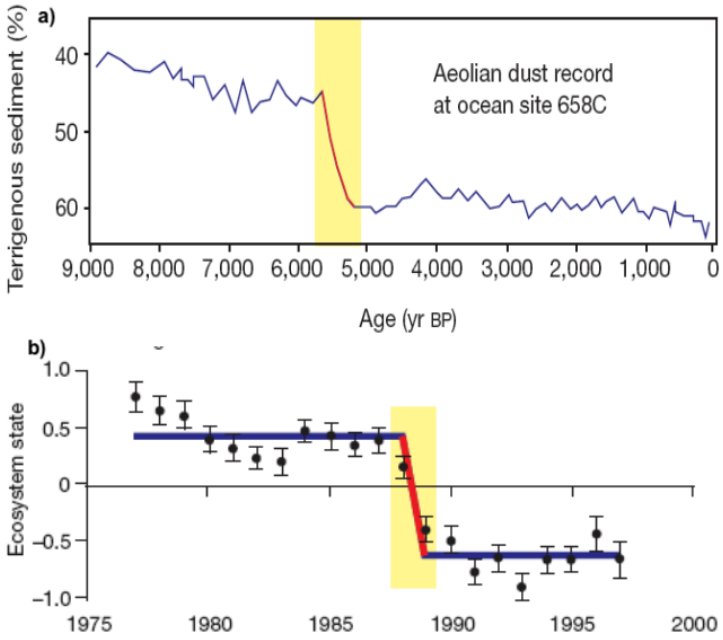


Figure 1.1. a) around 5,500 years ago, there was a sudden shift in the climate and vegetation of the Sahara, which is evident from the increased amount of land-based dust found in oceanic sediment; b) Regime shifts can also occur on much shorter timescales. For instance, an abrupt change in climatic and biological conditions was observed in the Pacific Ocean in 1989.

described in [15]: briefly, it consists of all those things that we can reach going a little further from our prior knowledge. The capacity to properly explore this space could discriminate innovation-prepared machines from inefficient ones.
Also, novelties and regime shift are strictly related to a long-standing problem of neural networks, that is the catastrophic forgetting [16]. It consist in the suddenly, complete erasing of past network's knowledge due to the learning of new tasks, as the name suggests. Paradoxically, this problem may originate from what makes neural network able to generalize so well, that say its single set of shared weights. The challenge is to design a network elastic enough to adapt and learn new input without damaging its prior knowledge, exactly what we tried to do.

What we said above about the optimal condition of the critical point could be not only a property of biological systems but also a general rule, as shown in [17], [2], [1], [13]: in similar manners, these authors showed that there is a relation between the edge of chaos, the frontier that divides the ordered regime from the chaotic one, and optimal performances on canonical datasets like MNIST, or text. Chaos theory was also used to better comprehend the learning process of neural networks, as in [4], and to make a network more prepared, through pre-training on a chaotic system, for natural phenomena datasets, as in [3].
So, differently from the examples above, we tried to actively bring the network to this edge of chaos during its training, in order to verify if in this configuration a neural network can better explore the Adjacent Possible, obtaining better reactivity

and performance on the sudden changes present in regime shifts.
The reader will be led through this work with four principal chapters: the present chapter, which will introduce all the basic concepts needed to understand the idea of the work, such as neural networks, regularizers, dynamical systems, and Lyapunov exponents; the second will explain the core idea of the work and its implementation; the third will show the applications of the idea and the results in various experimental contexts; the last is dedicated to the conclusions.

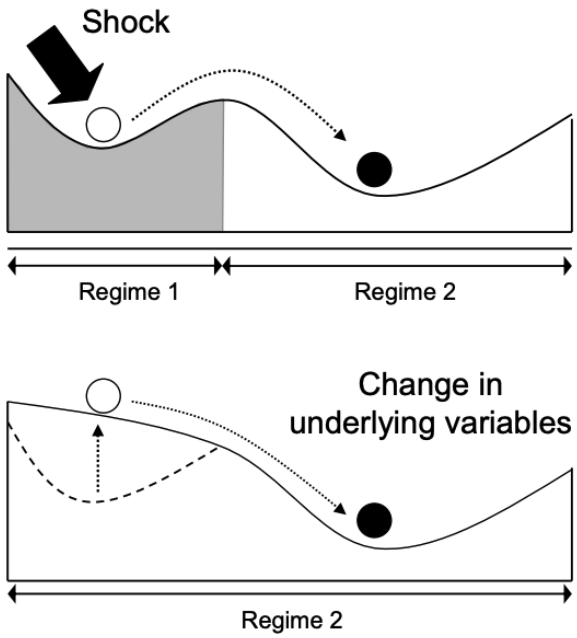


Figure 1.2. Regime shifts are typically caused by a combination of sudden shocks and gradual changes in underlying variables, along with internal feedbacks that alter the domains of attraction of different regimes. Slow changes in these underlying variables can result in the disappearance of some domains of attraction or the emergence of new ones that previously did not exist. The critical thresholds that separate different regimes are usually influenced by multiple underlying variables, rather than just a single factor.

1.1 AI, neural networks and deep learning

Artificial Intelligence (AI) refers to machines and algorithms capable of performing tasks typically handled by humans. Initially, AI successfully tackled intellectually challenging tasks that were simple for computers but difficult for humans. However, it struggled with tasks that humans do instinctively, such as recognizing speech or identifying faces, since its knowledge was hard-coded by humans.
The realization that AI systems need to learn autonomously from raw data led to the development of machine learning. Machine learning enables AI to handle real-world problems and make seemingly subjective decisions by identifying patterns in data. The effectiveness of these systems often depends on how data is represented. Representation learning, which allows the system to discover both the data representation and the mapping from input to output, has proven to enhance AI performance

significantly, and the quintessential of this concept is the multi-layer perceptron, the simplest neural network. It consists in a network of single interconnected units, the neurons, which, through a learning process, can solve specific tasks. Lastly, deep neural networks are networks of many successive layers of neurons which enables the algorithm to build progressively complex concepts out of simpler concepts (see Figure 1.5), magnify their representation ability.

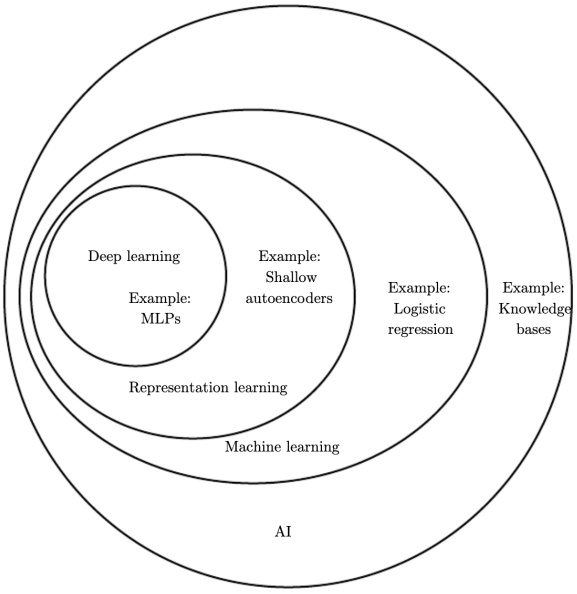


Figure 1.3. the major fields in AI.

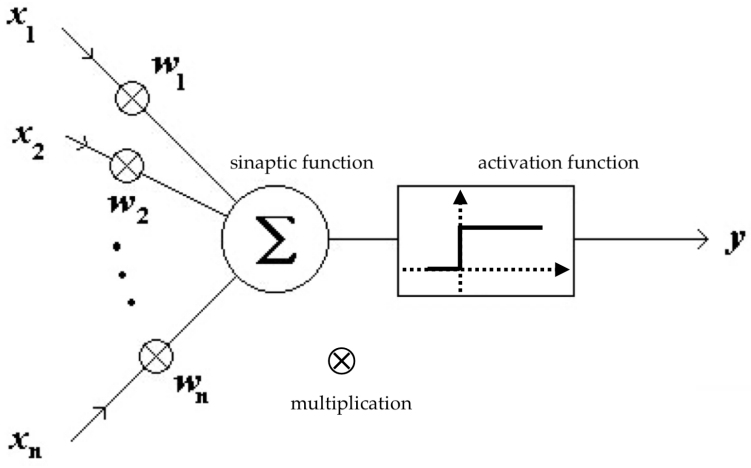


Figure 1.4. the simplest artificial neuron, the activation function is a non linear function and it could be of various type.

Similarly to how we learn, that is through the exposition to the reality that

surround us, the neural networks learn through the examples provided by us, and naturally the more are this examples the more will be the learning.
In practice a neurons is a linear combination of the input data whose result is the input of an activation function, as one can see in Figure 1.4:

$$f(\sum_i w_i x_i + b_i)$$

Where w_i are the connection weights between neurons, x_i are the inputs, b_i are the biases and f can take different forms, as one can see in Figure 1.6. Once the activation function is chosen, the network can start to learn a task by adjusting the strengths of the connections between neurons, a process called training. During training, the network take in input an example giving back an output, than the two are compared through a loss function, a function that measure how far they are. Given that the output depends on network's weights, the loss function too will depends on them. So, using the chain rule for partial derivatives and the back-propagation algorithm we can find its dependence from all the weighs of the network, thus modifying them accordingly to minimize the loss. The entire process is summarized in Figure 1.7.

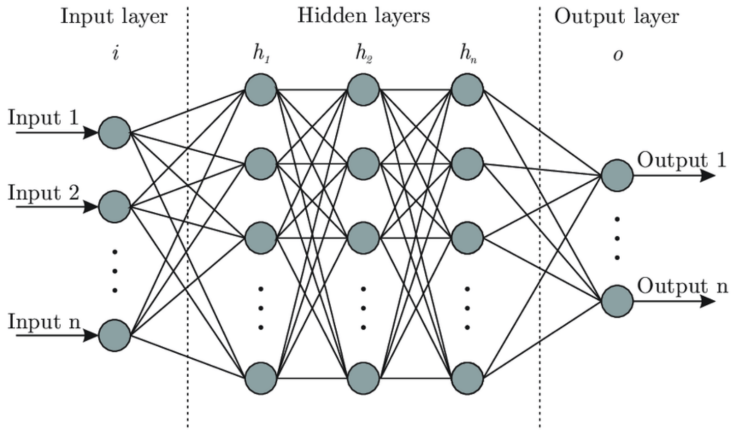


Figure 1.5. scheme of a deep neural network.

Although astonishing to see grow something pseudo-intelligent from nothing but algorithms, there are many problems with neural networks: the intrinsic unpredictability of this objects, that is the most recent developing still doesn't allow us to theoretically explain the entire learning process; the inability to understand syllogisms, the growing size of datasets, or the lack of elasticity in the paradigm shift.

As last remark for this section, we highlight that a neural network can be seen as a mathematical operator like $\hat{F} : \mathbb{R}^n \to \mathbb{R}^n$, rather than the combination of its single units. This point of view will be the keystone of the approach we will show in the next chapter.

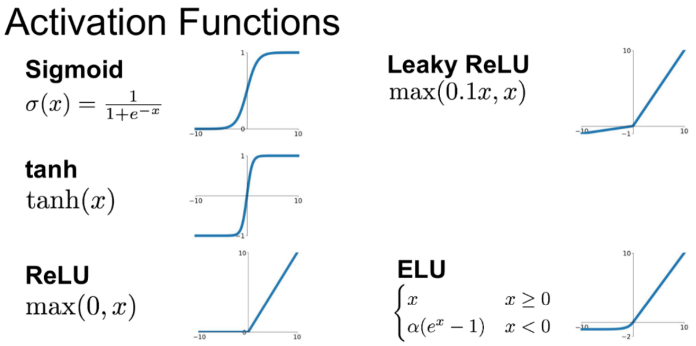


Figure 1.6. the most used activation functions.

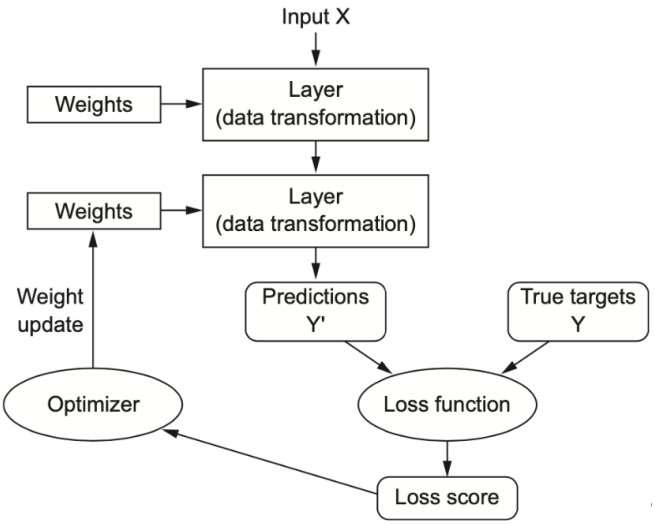


Figure 1.7. the general learning process in a network's training.

1.1.1 Open problems in machine learning

As mention before, neural networks and machine learning are far to be free from problems, indeed there are some deep criticality yet to be resolved, but we can hope to solve some of them:

- explainability and transparency
- generalization and overfitting
- continual learning and catastrophic forgetting

Now we are going to better explain these problem and the possible improvements we can offer.
Generalization refers to the ability of machine learning algorithm to make good predictions on unseen data, belonging to the same training data distribution. Is still

unclear why different type of neural network generalize better or not with respect others, or why very deep neural network, with a number of trainable parameter greater than data points, can still generalize well; but for sure, a network with more exploration ability can capture also rare events coming from the distribution tails of a certain phenomenon, improving the generalization where there is a lack of it. Also, a chaotic behavior, by virtue of its definition as we will see, can prevent the network from overfitting.

The problem of continual learning is still one of the big difference that divide actual neural networks from artificial general intelligence([23], [22]): it consists in the ability of continuously learning new tasks without forgetting how to perform the previous one, as a human being or an animal would do. The first attempts immediately showed a huge problem, the so called catastrophic forgetting. As the name suggest, it consist in the abrupt forgetting of a prior learned task few steps after the network begin to learn a new one. That the standard training process have some intrinsic criticality is a fact, indeed it was observed([16]) that the 'forgetting rate' of a certain task for a human being is very gradual during the learning of a new task, contrary to what happen in neural network. Moreover, if we think learning in terms of the weight-space, where a certain point in this space represent the optimal weights to perform a certain task, it's clear that perform well on a new task will require to move toward another weight configuration and so toward another point in weight-space. Now, if this space were smooth and predictable, catastrophic forgetting would not be catastrophic at all; but, as we know, for a neural network the landscape of this weight-space is much more complicated, reason why we see this phenomenon. In [23], the authors take inspiration directly from human brain to overcoming this problem: it was observed that in mouses' brain the volume of individual dendritic spines of neurons increases when an important task connected to these have to be remembered. Instead, spines associated to forgettable tasks progressively atrophy. This synapses' plasticity suggested to the authors a new mechanism to update the weights, that is a varying learning rate during training, in order to slow down learning on weights important to the tasks seen before.
This more 'mechanical' approach could be complementary to our 'dynamical' approach, more centered on modify the dynamics in the weight-space in an effective way.
But this approach of ours can touch also other problems, like explainability and vanishing/exploding gradient problem. Indeed, maybe neural networks are not anymore a black box, but surely they a gray one. We can't understand completely, in details, what happened during the whole process, and this is the reason why we can't solve some problems, like the ones showed above or the vanishing gradient. It often show up when dealing with long sequences of data, when the gradient, during back-propagation, grows uncontrollably or tends to zero, worsening the performance. In [19] and [20], besides providing some temporary solutions, the authors give a dynamical interpretation of the learning process, considering the network as an evolution map. In particular, they mathematically demonstrated that a simple network can't store information in a way robust to noise or outliers, while simultaneously having a non-vanishing gradient with respect to the initial inputs. In [20], in particular, a simulated annealing algorithm is suggested to mitigate the problem,

something that, in some way, resembles our chaotic exploration approach. So, a mathematical study of a chaotic neural network, besides to solve some practical problems, could explain some aspects of the learning process of neural networks in general, making them more intelligible.

1.1.2 Regularizers

Due to the central role in our idea, we are going to introduce the concept of regulizer(a detailed insight in [8]).
A key challenge in machine learning is creating an algorithm that performs well not only on the training data but also on new, unseen inputs. Many machine learning strategies are specifically designed to minimize test error, even if it means increasing the training error. These strategies are collectively referred to as regularization. To better understand this concept, we introduce some fundamental ideas: underfitting and overfitting, plastically represented in Figure 1.8. Underfitting occurs when a neural network is too simple to capture the underlying patterns in the data, and typically happens when the model hasn't enough neurons or layers, when the model is not trained long enough, when the input features are too few or not informative; overfitting is the opposite phenomenon, that is the network is too complex and tries too fit even the noise or the outliers.

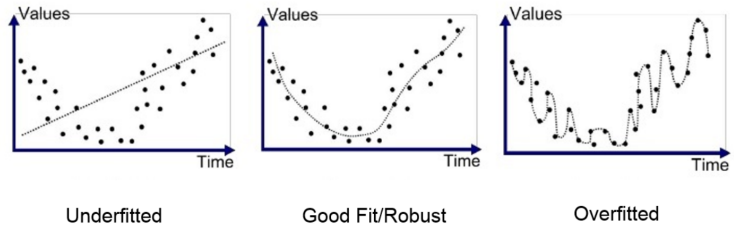


Figure 1.8. example of overfitting and underfitting, the regularizers' aim is to have a good fit.

Closely related to this concepts are those one of bias and variance (Figure 1.9). Bias regard the complexity level of the data' s underling structure: a model with high bias will oversimplifies or misleading the data structure, leading to underfitting; in case of low bias the model will be too flexible and will overfitting the data. Variance refers to the error introduced due to the model's sensitivity to small fluctuations in the training data. A model with high variance pays too much attention to the training data, including noise and outliers, leading to highly variable predictions.

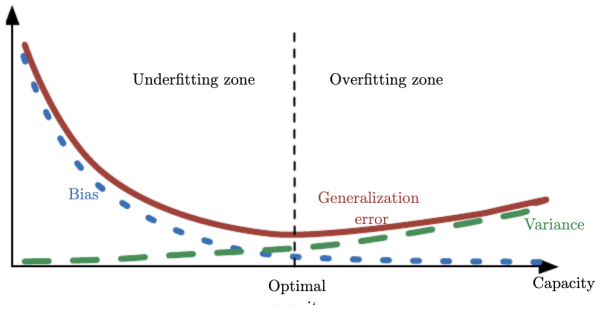


Figure 1.9. Generalization error as a function of capacity in the context of bias and variance; in loose words, a model's capacity is its ability to fit a wide variety of functions, and it grows with training.

So we use regularizers to address these problems, and there are many regularization strategies. Some of which impose additional constraints on the model, such as limiting the parameter values, others add extra terms to the objective function, which can be viewed as soft constraints on the parameters. When carefully selected, these constraints and penalties can enhance performance on the test set. In the context of deep learning, most regularization strategies are based on regularizing estimators, in which the network tries to make a profitable trade reducing variance significantly while not overly increasing the bias.

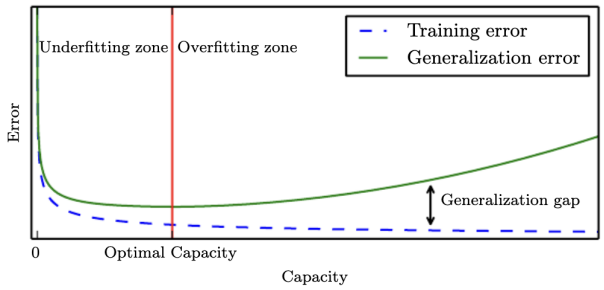


Figure 1.10. Training error and generalization error as a function of capacity.

The most used regularizers are:

- L^1: adds a penalty value to the loss function equal to the absolute value of the model parameters.
- L^2: regularization adds a penalty value equal to the squared value of the model parameters.
- Dropout: randomly drops some neurons from the model during training, which helps to avoid over reliance on a particular feature or combination of features.
- early stopping: consist in stopping the training process before its natural conclusion in order to save the weights which minimize the loss on the validation dataset.
- data augmentation: there many ways to implement it, but the main concepts is to slightly modify the input data in order to favor generalization.

In our case, the regularizer has the L^1 and L^2 characteristics, since it is a loss' additive term which exerts a soft constrain on the weights, but here this term not depends explicitly on them. So the loss function will be

$$Loss(y, \hat{y}) = d(y, \hat{y}) + R$$

where d is a certain distance measure and R is the new regularizer that will depends on the Lyapunov exponents of the network, in a way that will be clear in the following chapter.

1.2 Adjacent possible

This section is dedicated to an argument we already introduce, the Adjacent Possible. It was originally discussed by Stuart Kauffman in his molecular and biological investigations ([15]), but actually is something very general that can be applied in all fields of knowledge, as [12] demonstrated. It refers to the set of all things —ideas, molecules, technologies, etc.— that are one step away from what currently exists, a sort of edge of knowledge which hides all the further innovations too far for our comprehension. To better understand the previous statement we can think of it as a graph made up by connected node, as shown in Figure 1.11. As soon as a little step is done on this edge of knowledge, started from an isolated element or from a combination of connected elements, our sight expands on new possibilities and innovations, as one can see from the representation in Figure 1.11.

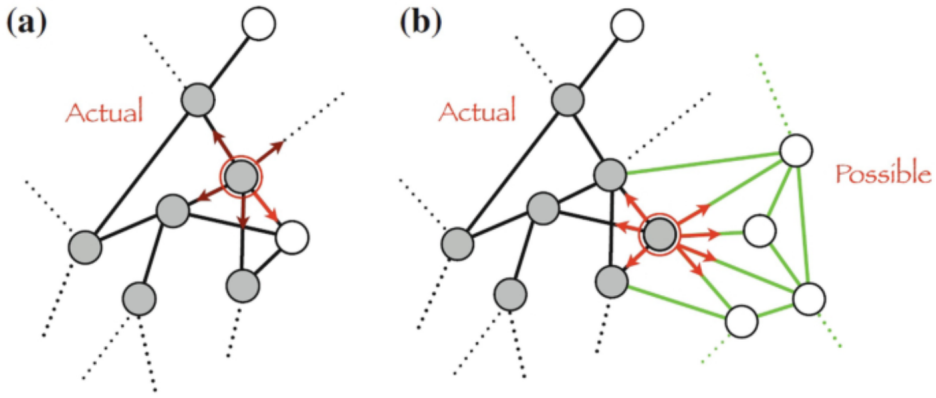


Figure 1.11. mathematical representation of adjacent possible through a graph.

But these are not only elusive speculation, indeed someone already found a quantitative approach to exploit the property of the adjacent possible, like one can see in [18], where the authors studied the curvature of the possible innovations space for different companies.
Moreover, the expansion in the adjacent possible, bringing innovations and so changes, could determine sudden regime shift in physical but also social phenomena, making very useful something that can quickly adapt to this changes.
So we want demonstrate that a system with chaotic properties is also able to better explore this adjacent possible.

1.3 Dynamical systems

Dynamical systems' category is really wide and in general it includes all the phenomena evolving in time that we are able to describe through differential equations. In other words, we define dynamical system any mathematical model or rule which determines the future evolution of the variables describing the state of the system, from their initial values.
We have been studying systems from a dynamical point of view since Newton opened this path in the mid-1600s, with the three laws of dynamics, a formalism useful for many classical problem but impracticable when the degrees of freedom arise or nonlinarity appears: an example is the planets' motion, easy explainable for two corpses but impossible to solve for three. Despite the impossibility to know the exact solution of some problems, is still possible to discover interesting behavior of such systems using a geometrical approach, like the one utilized by Poincaré in the 1800s and pursued by many mathematician and physicists in the years that followed. In fact, only looking at the equations' structure, we can obtain the dynamic of the system in the phase space, and so the qualitative behavior. In particular, a dynamical system can be expressed by a certain number of differential equation like these

$$\begin{aligned} \frac{\mathrm{d}x_1}{\mathrm{d}t} &= f_1(x_1(t), x_2(t), \cdots, x_d(t)) \\ &\cdot \\ &\cdot \\ &\cdot \\ \frac{\mathrm{d}x_d}{\mathrm{d}t} &= f_d(x_1(t), x_2(t), \cdots, x_d(t)). \end{aligned} \tag{1.1}$$

So, at any given time t, the state of a generic system is determined by the values of all variables which specify its state of motion, i.e. $x(t) = (x_1(t), x_2(t), x_3(t), ..., x_d(t))$, d being the system dimension. The set of all possible states of the system, i.e. the allowed values of the variables $x_i (i = 1, ..., d)$, defines the phase space of the system. More precisely, 1.1 defines autonomous ordinary differential equation as the functions f_i do not depends on time. If the functions also depend on time, the system is considered non-autonomous, making the problem even more complex. The most commonly studied systems are smooth dynamical systems, which involve differentiable functions and for which the theorem of existence and uniqueness applies. This theorem guarantees that there is a unique solution $x(t)$ to the differential equation 1.1, as long as the initial condition $x(0)$ is specified. However, this does not necessarily mean that the trajectory $x(t)$ can be practically predicted.
An example is the pendulum in his small angle approximation:

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} + \omega_0^2 \sin\theta \approx \frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} + \omega_0^2\theta = 0$$

that, choosing $\theta = x_1$ and $\dot{\theta} = x_2$, can be rewrite as

$$\begin{aligned} \frac{dx_1}{dt} &= x_2 \\ \frac{dx_2}{dt} &= -\omega_0^2 x_1 \end{aligned} \tag{1.2}$$

giving the phase space trajectory in Figure 1.12.

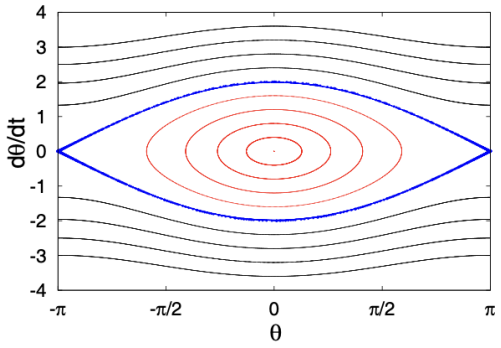


Figure 1.12. pendulum in linear approximation in the phase space; red lines represent oscillations, the blue one is the separatrix, and the black ones are the rotations.

We can identify two general classes of dynamical systems. To introduce them, let's imagine to have N pendulums and to choose a slightly different initial state for any of them. Now put all representative points in phase space Γ forming an ensemble whose distribution is described by a probability density function $\rho(x, t = 0)$ normalized in such a way that $\int_\Gamma dx\rho(x, 0) = 1$. The number of pendulums cannot change so that $dN/dt = 0$. The latter result can be expressed via the continuity equation

$$\frac{\partial \rho}{\partial t} + \sum_{i=1}^{d} \frac{\partial f_i \rho}{\partial x_i} = 0$$

where $\rho\mathbf{f}$ is the flux of representative points in a volume $d\mathbf{x}$ around $\mathbf{x}$. The above equation can be rewritten as

$$\partial_t \rho + \sum_{i=1}^{d} f_i \partial_i \rho + \rho \sum_{i=1}^{d} \partial_i f_i = \partial_t \rho + \mathbf{f} \cdot \nabla \rho + \rho \nabla \cdot \mathbf{f} = 0 \tag{1.3}$$

where $\partial_t = \partial/\partial t$ and $\nabla = (\partial_1, ..., \partial_d)$. We can now distinguish two classes of systems depending on the vanishing or not of the divergence $\nabla \cdot \mathbf{f}$: If $\nabla \cdot \mathbf{f} = 0$, the velocity field $\mathbf{f}$ is uncompressible and the phase-space volumes are conserved; we thus speak of conservative dynamical systems. If $\nabla \cdot \mathbf{f} < 0$, phase-space volumes contract and we speak of dissipative dynamical systems.

For example, in the presence of friction we have that $\nabla \cdot \mathbf{f} < 0$ and, if an external force is absent, the whole phase space contracts to a single point. In general, the set of points asymptotically reached by the trajectories of dissipative systems lives in a space of dimension $D < d$, and such a set is called attractor.

These are the basic concepts to comprehend the dynamical system that surround us and that we are going to see in the next sections, like the Lorenz system.

1.3.1 Stability and bifurcation

Since Lorenz system present bifurcation phenomena, we shall do a brief introduction on them too(more details in [6] ans [5]).
So, we mentioned the fact that the solutions of a differential equation could be visualized as trajectories flowing through an d-dimensional phase space with coordinates $(x_1, ..., x_d)$. It's easier to introduce some concepts, which will be useful later, in a one-dimensional space.
For example, if we consider the differential equation $\dot{x} = f(x)$, it represents a vector field on the line: it dictates the velocity vector $\dot{x}$ at each x.

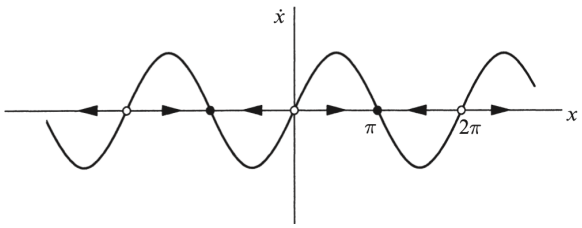
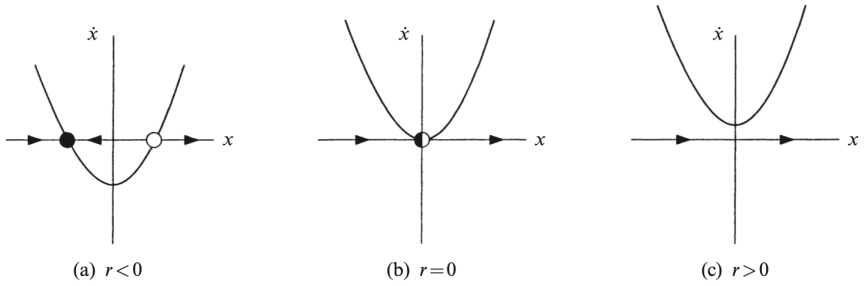


Figure 1.13. example of one-dimensional flow, in particular it describes the differential equation $\dot{x} = \sin(x)$

The flow is to the right when $\dot{x} > 0$ and to the left when $\dot{x} < 0$, as shown in Figure 1.13; instead, at points where $\dot{x} = 0$, the fixed points ,there is no flow. You can see that there are two kinds of fixed points in Figure 1.13: black dots represent stable fixed points and open circles represent unstable fixed points. I terms of equilibrium, stable fixed points represents stable equilibria, as all sufficiently small perturbation dump out in time, while unstable fixed points represents unstable equilibria, as perturbations grows in time.
There are some cases, especially in the context of bifurcations, in which the fixed point is not stable nor unstable, but half-stable, since it's attracting from the left and repelling from the right.

Now we can introduce the bifurcation: all the transition in which fixed points are created, destroyed or change their stability varying the system parameters, are called bifurcations, and the parameter values at which they occur are called bifurcation points. Bifurcations are scientifically important since they provide models of transitions and instabilities as some control parameter is varied. In the following we will briefly introduce the main types of bifurcations: saddle-node bifurcation, transcritical bifurcation and pitchfork bifurcation.
Saddle-node bifurcation is the basic mechanism by which fixed points are created and destroyed. As a parameter is varied, two fixed points move toward each other, collide, and mutually annihilate, as shown in Figure 1.14 and Figure 1.15.



(a) $r<0$ (b) $r=0$ (c) $r>0$

Figure 1.14. the basic example of a saddle-node bifurcation in the system $\dot{x} = r + x^2$

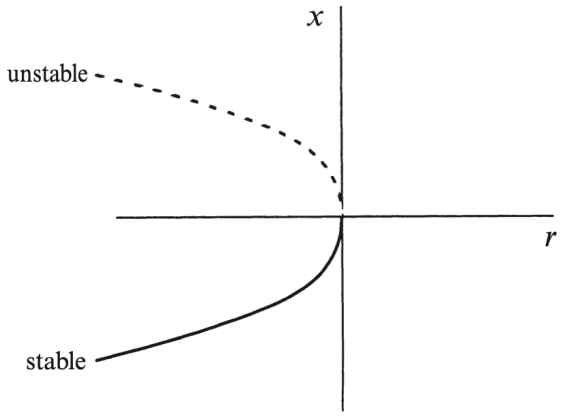
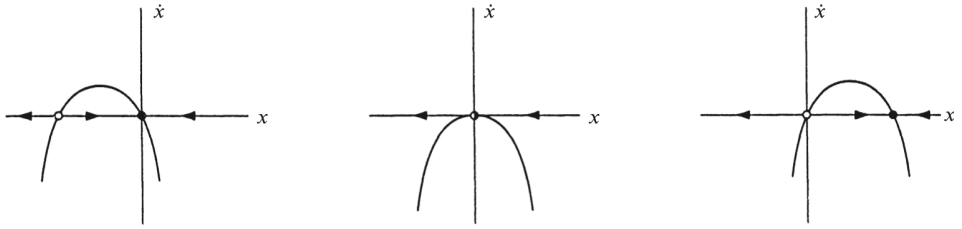


Figure 1.15. bifurcation diagram for the saddle-node bifurcation in one dimension for the differential equation $\dot{x} = r + x^2$.

Instead, when a fixed point exists for all values of the parameter only changing its stability, we are in presence of a transcritical bifurcation, as shown in Figure 1.16 and Figure 1.17.



(a) $r<0$ (b) $r=0$ (c) $r>0$

Figure 1.16. transcritical bifurcation in the differential equation $\dot{x} = rx - x^2$.

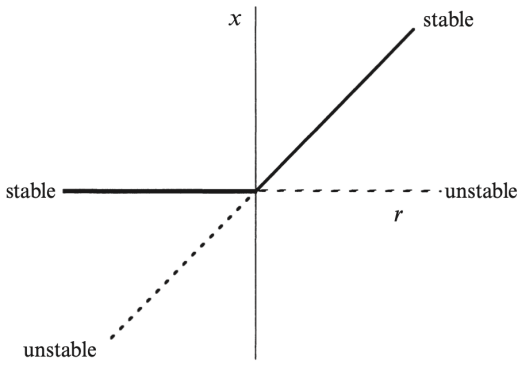


Figure 1.17. bifurcation diagram for the transcritical bifurcation in one dimension for the equation $\dot{x} = rx - x^2$.

We turn now to the third kind of bifurcation, the so-called pitchfork bifurcation. This bifurcation is common in physical problems that have a symmetry. In such cases, fixed points tend to appear and disappear in symmetrical pairs. There are two very different types of pitchfork bifurcation, supercritical and subcritical, showed in Figure 1.18.

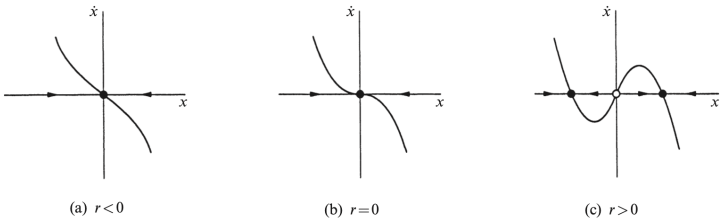


Figure 1.18. supercritical bifurcation for the differential equation $\dot{x} = rx - x^3$; the subcritical bifurcation observable in $\dot{x} = rx + x^3$ the diagrams are mirror.

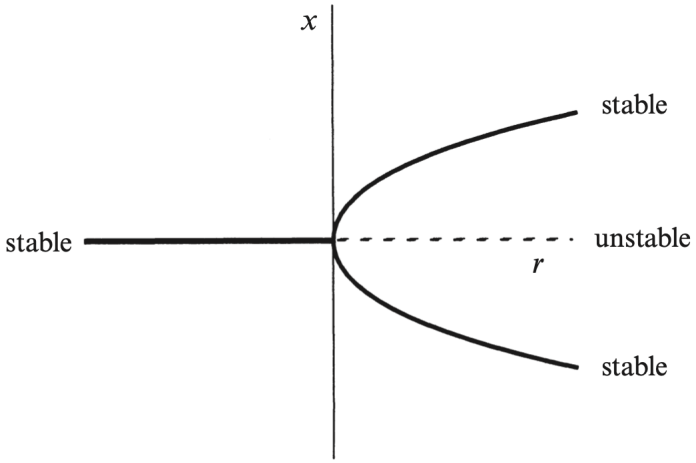


Figure 1.19. bifurcation diagram for a supercritical pitchfork bifurcation in one dimension.

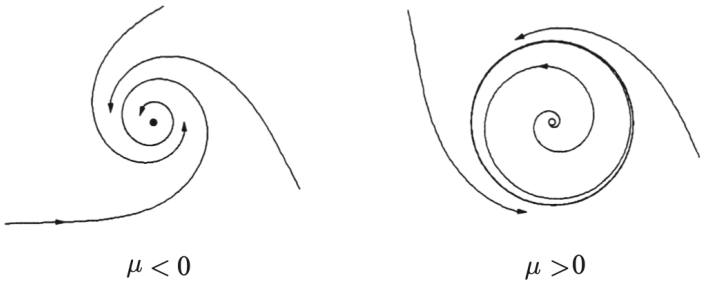


Figure 1.21. example of supercritical Hopf bifurcation.

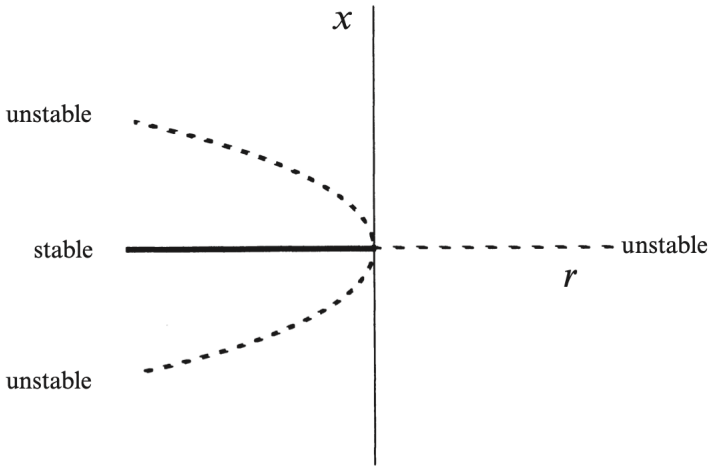


Figure 1.20. bifurcation diagram for a subcritical pitchfork bifurcation in one dimension.

We conclude the phenomenology present in the Lorenz system with a brief, intuitive, explanation of Hopf bifurcation.
One can observe it in 2-dimensional systems or more and presents a supercritical and subcritical variant. In terms of flow in phase space, a supercritical Hopf bifurcation occurs when a stable spiral goes into an unstable spiral, which is then encircled by a small limit cycle, like in 1.21.

Like pitchfork bifurcations, Hopf bifurcations can be either supercritical or subcritical. The subcritical variety is generally more dramatic and can be potentially dangerous in engineering applications. After a subcritical bifurcation, the system's trajectories must jump to a distant attractor, which could be a fixed point, another limit cycle, infinity, or—in systems with three or more dimensions—a chaotic attractor. The phase portraits for this scenario is illustrated in 1.22. When $\mu < 0$, there are two attractors: a stable limit cycle and a stable fixed point at the origin, with an unstable cycle lying between them. As μ increases, the unstable cycle contracts around the fixed point. A subcritical Hopf bifurcation occurs at $\mu = 0$, where the unstable cycle shrinks to zero amplitude and engulfs the origin, making it unstable. For $\mu > 0$, the large-amplitude limit cycle becomes the only attractor, forcing solutions that once stayed near the origin to expand into large-amplitude

oscillations.

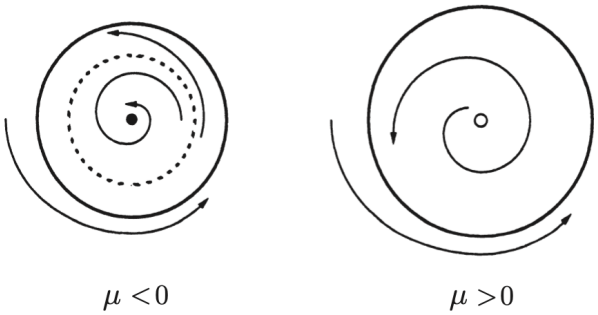


Figure 1.22. example of subcritical Hopf bifurcation.

1.3.2 Lorenz system

The dynamical system we will use in the following, one of the most known systems, is defined by the Lorenz equations, investigated by Lorenz in the '60s during his studies on turbulent fluids:

$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= rx - y - xz \\ \dot{z} &= xy - bz \end{aligned} \tag{1.4}$$

With this system the concept of chaos break into many fields of knowledge: it was the first example of something completely deterministic but at same time with unpredictable characteristics; moreover the same equations also arise in models of lasers and dynamos, but also in the motion of a certain waterwheel.
Let's begin describing some simple properties of this system until we make sense of the Figure 1.23.

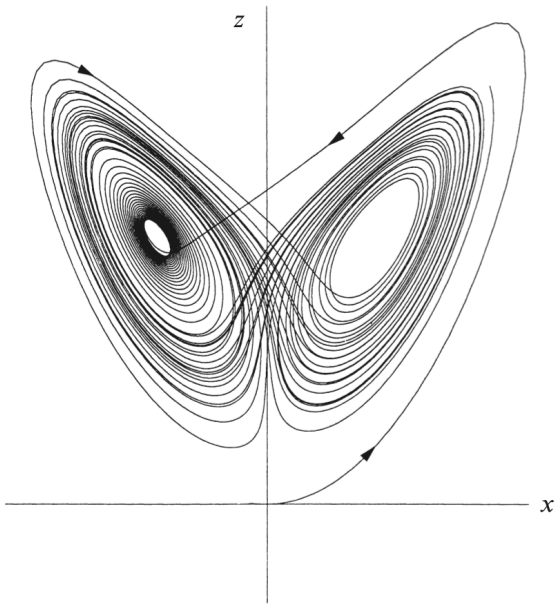


Figure 1.23. projection of Lorenz attractor on xz plane, with parameters $\sigma = 10$, $b = 8/3$, $r = 28$.

It's easy to show that for Lorentz systems the volume in phase space contract under the flow:

$$\nabla \cdot \mathbf{f} = \frac{\partial}{\partial x}[\sigma(y-x)] + \frac{\partial}{\partial y}[rx - y - xz] + \frac{\partial}{\partial z}[xy - bz] = -\sigma - 1 - b < 0.$$

Since the divergence is constant, we have $\dot{V} = (-\sigma - 1 - b)V$, which has solution $V(t) = V(0)e^{(-\sigma-1-b)t}$. Thus volumes in phase space shrink exponentially fast. Hence, if we start with an enormous solid blob of initial conditions, it eventually shrinks to a limiting set of zero volume.
In fact, looking at the Lorenz equations one can see that there is 3 fixed points for the dynamics: one is the origin, for all values of the parameters; the other two are a symmetric pair of fixed points $x^* = y^* = \pm\sqrt{b(r-1)}$, $z^* = r - 1$ that Lorenz called C^+ and C^-, but they exist only for r>1. As $r \to 1^+$, C^+ and C^- coalesce with the origin in a pitchfork bifurcation. Is possible to demonstrate that the origin is a saddle point for r>1 and globaly stable for r<1, hence there can be no limit cycle or chaos.
In the case of $r > 1$, C^+ and C^- are linearly stable for

$$1 < r < r_H = \frac{\sigma(\sigma + b + 3)}{\sigma - b - 1}$$

(assuming also that $\sigma - b - 1 > 0$), and lose stability at $r = r_H$ in a Hopf bifurcation. Given what we said in previous section, one could think that, for r slightly greater than r_H, a stable limit cycle would appear around C^+ and C^-. But it isn't the case because it was demonstrated that this Hopf bifurcation is subcritical, and so the limit cycles are unstable and exist only for $r < r_H$.

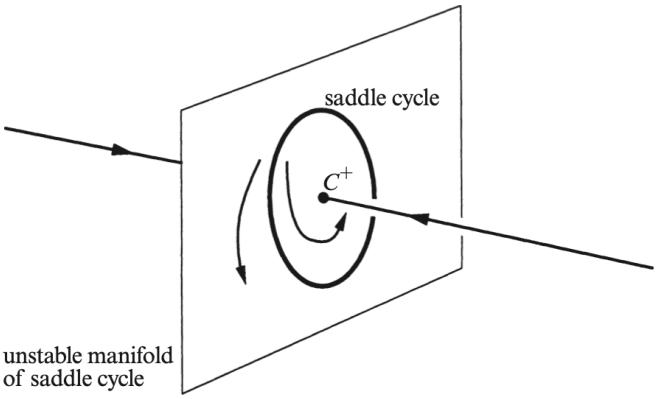


Figure 1.24. The phase portrait near C^+ for $r > r_H$.

So the stable fixed points are encircled by a saddle cycle, a type of unstable limit cycle that is possible only in phase spaces of three or more dimensions, shown in Figure 1.24. As $r \to r_H$ from below, the cycle shrinks around the fixed point, and at the bifurcation it absorbs the saddle cycle and changes into a saddle point. For $r > r_H$ there are no attractors in the neighborhood of C^+ and C^-, so the trajectories must fly away to a distant attractor. A partial bifurcation diagram for the system, based on the results so far, shows no hint of any stable objects for $r > r_H$, as one can see in 1.25.

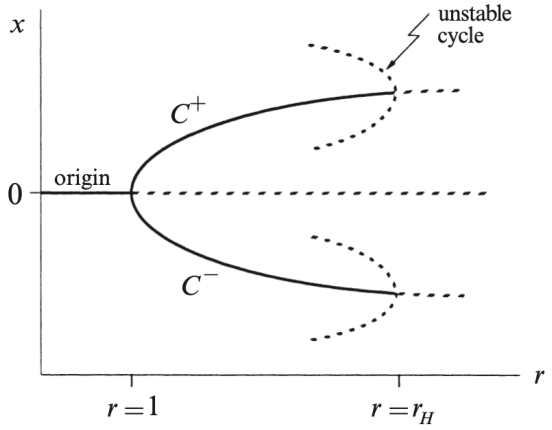


Figure 1.25. Partial bifurcation diagram for the Lorenz system.

At the same time, trajectories can't go to infinity due to the dissipative nature of Lorenz system; moreover Lorenz gave an argument that for r slightly greater than r_H, any limit cycles would have to be unstable. So the only possibility is that the trajectory go to one unstable object to another without intersections and simultaneously they are confined in a set of zero volume.
Now Figure 1.23 is more comprehensible. From Figure 1.26 is even clearer how, after an initial transient, the solution settles into an irregular oscillation that persists as $t \to \infty$, but never repeats exactly, that say the motion is aperiodic.

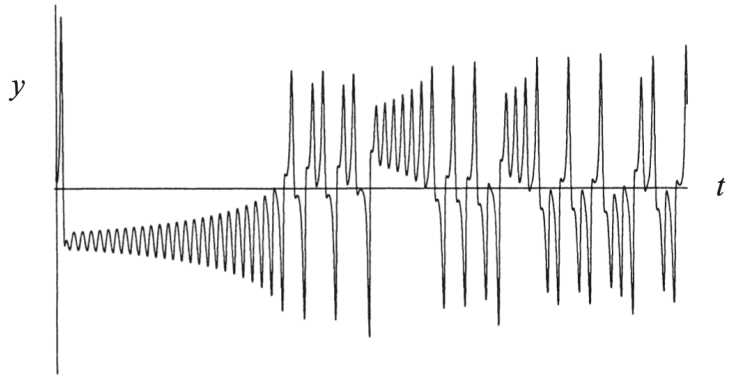


Figure 1.26. Plot of y(t) of the numerical solution of Lorenz system.

It can seems that the Lorenz strange attractor is made up of two merging surfaces, but in reality is a fractal, something with zero volume and infinite surface area. Lorenz described the attractor in this way:

"It would seem, then, that the two surfaces merely appear to merge, and remain distinct surfaces. Following these surfaces along a path parallel to a trajectory, and circling C^+ and C^-, we see that each surface is really a pair of surfaces, so that, where they appear to merge, there are really four surfaces. Continuing this process for another circuit, we see that there are really eight surfaces, etc., and we finally conclude that there is an infinite complex of surfaces, each extremely close to one or the other of two merging surfaces."

Until now we discovered the canonical Lorenz strange attractor observing only a small interval of r values, but there is an entire three-dimensional parameter space to be explored, and much remains to be discovered. In the following we explore a little more the parameter space maintaining $\sigma = 10, b = 8/3$ while varying r, in order to observe some interesting phenomena.
As we decrease r from r_H , the unstable limit cycles expand until, at $r \approx 13.93$ it touches the saddle point and become an homoclinic orbits -trajectories that start and end at the same fixed point are called homoclinic orbits; notice that a homoclinic orbit does not correspond to a periodic solution, because the trajectory takes forever trying to reach the fixed point-. Hence we have a homoclinic bifurcation and below $r = 13.93$ there are no limit cycles. The main conclusion is that an invariant set is born at $r = 13.93$, along with the unstable limit cycles. It is not an attractor and is not observable directly, but it generates sensitive dependence on initial conditions in its neighborhood. One can observe trajectories that initially go chaotically around until they settle down to C^+ or C^-. The time spent wandering near the set gets longer and longer as r increases. Finally, at $r = 24.06$ the time spent wandering becomes infinite and the set becomes a strange attractor.

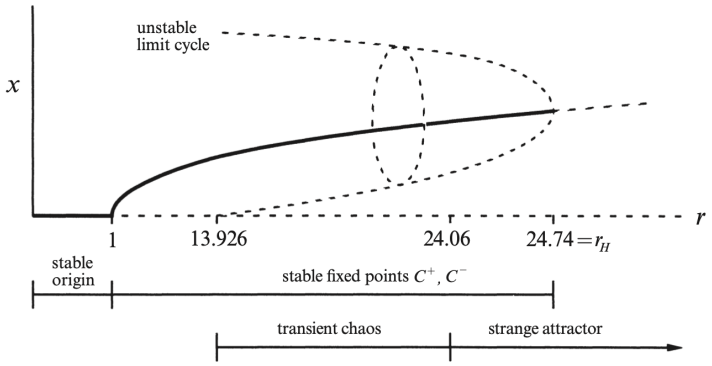


Figure 1.27. a schematic view of the system behavior varying the parameter r.

The phenomenon described above for r in the interval [13.93, 24.06] is called transient chaos, shown in 1.28, and it shows that a deterministic system can be unpredictable, even if its final states are very simple. In particular, you don't need strange attractors to generate effectively random behavior. We note one other interesting implication of 1.27: for $24.06 < r < 24.74$, there are two types of attractors: fixed points and a strange attractor. This coexistence means that we can have hysteresis between chaos and equilibrium by varying r slowly back and forth.

Transient chaos and hysteresis are examples of the those critical phenomena or regime shifts we talked about in the first chapter: phenomena at first sight chaotic can suddenly collapse in an equilibrium regime, while other systems can dramatically pass from order phase to chaotic phase due to an external force changing its parameters.

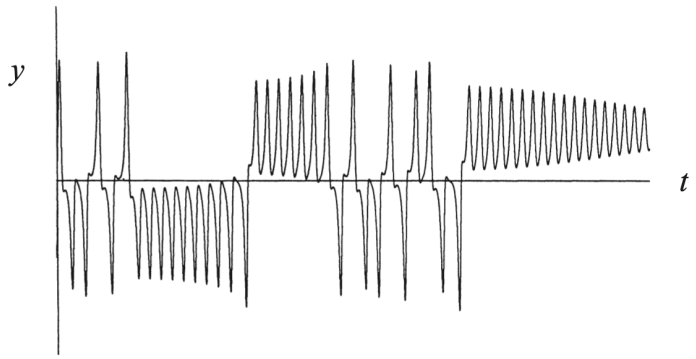


Figure 1.28. y-projection of Lorenz system solution in the case of transient chaos.

1.3.3 Lyapunov exponents

We said that strange attractor exhibits sensitive dependence on initial conditions: this means that two trajectory starting very close together will rapidly diverge from each other. For example, if we have as initial conditions $X(t)$ and $X(t) + \delta(t)$, it is well known that on Lorenz attractor δ grows exponentially: $||\delta(t)|| \sim ||\delta_0||e^{\lambda t}$, as

one can see in Figure 1.29 and 1.31. The rate λ of this divergence is the Lyapunov exponent.

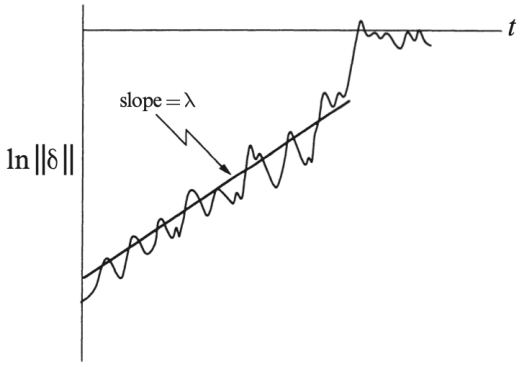


Figure 1.29. estimate of the exponential divergence of two near trajectories in chaotic systems.

Looking at Figure 1.29, we notice that the curve fluctuate and reach a saturation: the first phenomenon is due to varying strength of the exponential along the trajectory; the second due too the fact that trajectories can't go further than attractor diameter. We have also to specify that, in this naive example, what we call λ is only one of a set of Lyapunov exponents, the maximum one. Indeed, in an n-dimensional system, there will be n different Lyapunov exponents, each of them control the trajectory expansion or contraction on a certain space direction. But for long time, the behavior of the system is controlled by the maximum Lyapunov exponent. Moreover, λ slightly depends on trajectories, so one should average over many different points on the same trajectory to get the true value of λ.
Given this sensitive dependence on initial conditions, making predictions based on data from chaotic systems is very hard. This is because, due to the intrinsic uncertainty in the measuring process, all the predictions that one can make have a finite validity horizon, after which the difference between prediction and real data overcome our measure tolerance. For example if c is the tolerance, the prediction will fail when $||\delta(t)|| \geq c$, that is

$$t_{horizon} = O(\frac{1}{\lambda}\frac{c}{||\delta_0||})$$

Independently of our efforts, until there will be a minimal difference between measure and reality, we won't be able to predict longer than a few multiples of $1/\lambda$.

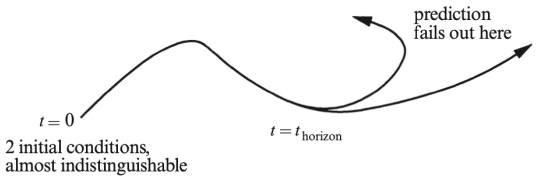


Figure 1.30. divergence of prediction and real data.

As we said, in a multidimensional system there are d Lyapunov exponents, but we didn't define them formally; to do so we consider a d dimensional map:

$$\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t))$$

Lyapunov spectrum can be computed by studying the time-growth of d independent infinitesimal perturbations $[\mathbf{w}^{(i)}]_{i=1}^{d}$ with respect to a reference trajectory. In mathematical language, the vectors $\mathbf{w}^{(i)}$ span a linear space, the tangent space. The evolution of a generic tangent vector is obtained by linearizing the previous map:

$$\mathbf{w}(t+1) = \mathbb{L}[\mathbf{x}(t)]\mathbf{w}(t) \tag{1.5}$$

where $\mathbb{L}_{ij}[\mathbf{x}(t)] = \partial f_i(\mathbf{x})/\partial x_j|_{\mathbf{x}(t)}$ is the linear stability matrix or jacobian. From eq. 1.5 we see that the stability problem consists in studying the asymptotic properties of a matrix product, indeed we can write $\mathbf{w}(t)$ in terms of the initial condition $\mathbf{x}(0)$ and $\mathbf{w}(0)$ as $\mathbf{w}(t) = \mathbb{P}_t[\mathbf{x}(0)]\mathbf{w}(0)$, where

$$\mathbb{P}_t = \prod_{k=0}^{t-1} \mathbb{L}[\mathbf{x}(k)]$$

In this context, a result of particular relevance is provided by Oseledec (1968) multiplicative theorem

Theorem 1 (Osedelec theorem)**.** *Let $\mathbb{L}(1), \mathbb{L}(2), ..., \mathbb{L}(k), ...$ be a sequence of $d \times d$ stability matrices referring to the evolution rule given by the map, assumed to be an application of the compact manifold A onto itself, with continuous derivatives. Moreover, let μ be an invariant measure on A under the evolution of the same map. The matrix product $\mathbb{P}_t[\mathbf{x}(0)]$ is such that, the limit*

$$\lim_{t\to\infty} \left[\mathbb{P}_t^T[\mathbf{x}(0)]\mathbb{P}_t[\mathbf{x}(0)]\right]^{\frac{1}{2t}} = \mathbb{V}[\mathbf{x}(0)]$$

exists with the exception of a subset of initial conditions of zero measure. Where $\mathbb{P}^T$ denotes the transpose of $\mathbb{P}$.

The symmetric matrix $\mathbb{V}[\mathbf{x}(0)]$ has d real and positive eigenvalues $\nu_i[\mathbf{x}(0)]$ whose logarithm defines the Lyapunov exponents

$$\lambda_i(\mathbf{x}(0)) = \ln(\nu_i[\mathbf{x}(0)]).$$

Customarily, they are listed in descending order $\lambda_{max} = \lambda_1 \geq \lambda_2 .... \geq \lambda_d$, equal sign accounts for multiplicity due to a possible eigenvalue degeneracy. The Oseledec theorem guarantees the existence of Lyapunov exponents for a broad range of dynamical systems under very general conditions. As mentioned earlier, Lyapunov exponents are linked to individual trajectories, which means that we cannot ignore the dependence on the initial condition $x(0)$ unless the dynamics is ergodic. In an ergodic system, the Lyapunov spectrum becomes independent of the initial condition, turning into a global property of the system.
A consequence of the Oseledec theorem regards the expansion rate of k-dimensional

oriented volumes $Vol_k(t) = Vol[w^{(1)}(t), w^{(2)}(t), ..., w^{(k)}(t)]$ delimited by k independent tangent vectors $w^{(1)}, w^{(2)}, ..., w^{(k)}$. Under the dynamics flow, the k-parallelepiped is distorted and its volume-rate of expansion/contraction is given by the sum of the first k Lyapunov exponents:

$$\sum_{i=0}^{k} \lambda_i = \lim_{t\to\infty} \frac{1}{t} \ln\left(\frac{Vol_k(t)}{Vol_k(0)}\right) \tag{1.6}$$

When we consider k-volumes with k = d, d being the phase-space dimensionality, the sum 1.6 gives the phase-space contraction rate,

$$\sum_{i=1}^{d} \lambda_i = \langle \ln |\det[\mathbb{L}(x)]| \rangle,$$

which for continuous time dynamical systems reads

$$\sum_{i=1}^{d} \lambda_i = \langle \nabla \cdot f(x) \rangle,$$

angular brackets indicates time average. Therefore, recalling the distinction between conservative and dissipative dynamical systems, we have that for the former the Lyapunov spectrum sums to zero.

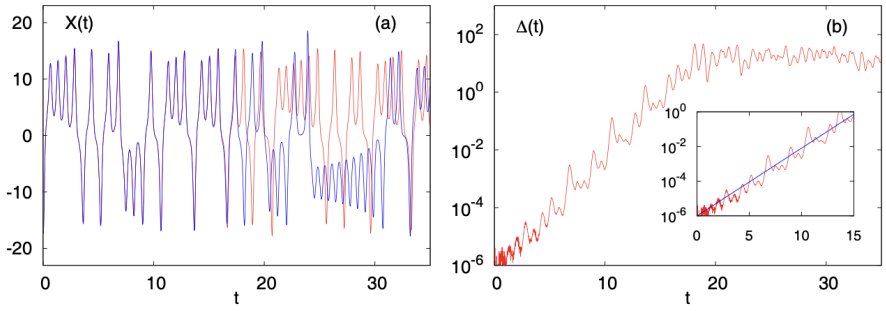


Figure 1.31. Lyapunov exponent Lorenz model: (a) evolution of reference $X(t)$ (red) and perturbed $X'(t)$ (blue) trajectories, initially at distance $\Delta(0) = 10^{-6}$. (b) Evolution of the separation between the two trajectories. Inset: zoom in the range $0 < t < 15$ in semi-log scale.

The important characteristic of Lyapunov exponents is their capacity to reveal the system behavior. Indeed, in the case of dissipative systems, the set of Lyapunov exponents is informative about qualitative features of the attractor. For example, if the attractor reduces to

- stable fixed point, all the exponents are negative;
- limit cycle, an exponent is zero and all the other are negative;
- k-dimensional stable torus, the first k Lyapunov exponents vanish and all the others are negative;

- for strange attractor generated by a chaotic dynamics at least one exponent is positive.

The concepts illustrated in this section, highlights a great problematic in the prediction at least of the chaotic phenomena. Because, besides the difficulties due to nonlinearities and aperiodic motions, the sensitive dependence on initial conditions clashes with our finite possibilities, in physical measuring and in computer simulations. Indeed, even a difference of some bits, not only in the initial condition but also in the control parameter, will be evident if we wait enough, making us wonder if what we see on computer is even real. For example, in Figure 1.32, we illustrate four different trajectories of the Lorenz equations, each obtained by introducing an infinitesimal error in comparison to a reference trajectory. This error could be in the initial condition, the integration step, or the model parameters. The effect of these errors, regardless of their source, is similar: all trajectories remain nearly identical for a while before diverging significantly after a certain period, which is determined by the initial deviations from the reference trajectory or system. This example highlights that sensitivity in chaotic systems is not only dependent on initial conditions but also on the evolution laws and the algorithmic implementation of the models.
So it's natural wonder about the feasibility of using such systems to model natural phenomena and the impact of chaos on experiments conducted in both laboratory settings and computer simulations. But fortunately there is a theorem, Tucker's theorem, reassuring us that, despite these inevitable numerical errors, the strange attractor and the chaotic motion that we see are genuine properties of the Lorenz equations themselves. So, the qualitative phenomena we investigated through neural networks, remain something useful to evaluate the results obtained here and the ones that could be obtained on a real system.

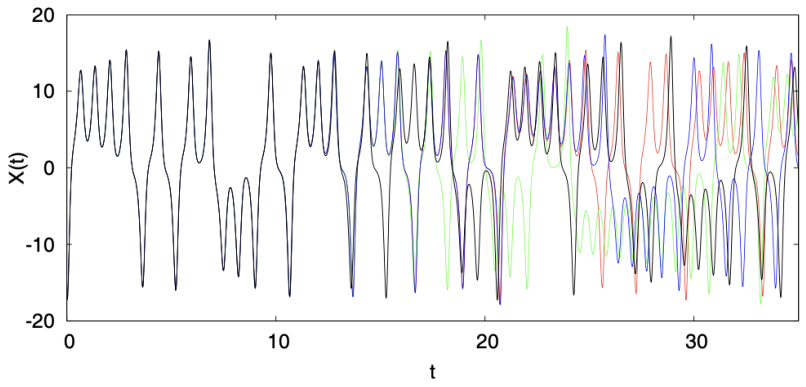


Figure 1.32. $X(t)$ versus time for the Lorenz model at $r = 28, \sigma = 10$ and $b = 8/3$: in red the reference trajectory, in green that obtained by displacing of an infinitesimal amount the initial condition, in blue by a tiny change in the integration step with the same initial condition as in the reference trajectory, in black evolution of same initial condition of the red one but with r perturbed by a tiny amount.

1.3.4 Algorithm for computing Lyapunov Spectrum

In this section we illustrate the computation methods used in the implementation of the regularizer (more details in [7] and [9]).
In one case, we applied directly the Osedelec theorem computing the jacobians and the matrix product, but obviously we can't perform the limit for $t \to \infty$, and so this method is only an approximation. Although this practical problem, is still possible to adopt this solution with good result, but the training may results unstable. The implementation is straight forward, as it requires the computations prescribed by the theorem, entrusting the differentiation to the computer.

In the second case, we use a more efficient algorithm, the recursive QR decomposition. The quantities one wishes to compute are the eigenvalues of the Oseledec matrix:

$$\mathbb{P}_t^T[\mathbf{x}(0)]\mathbb{P}_t[\mathbf{x}(0)] = \mathbb{L}(1)^T \cdot \mathbb{L}(2)^T, ..., \mathbb{L}(t)^T \cdot \mathbb{L}(t) \cdot \mathbb{L}(t-1), ..., \mathbb{L}(1) \tag{1.7}$$

but the problem is that this matrix is ill-conditioned for large t. The coefficient matrix is called ill-conditioned when a small change in the constant coefficients results in a large change in the solution, so in the presence of rounding errors, ill-conditioned systems are inherently difficult to handle. But the QR decomposition resolve the problem by effectively partially diagonalizing the matrix step by step.
So, we saw that Osedelec matrix is present the matrix multiplication $\mathbb{L}(t) \cdot \mathbb{L}(t-1), ..., \mathbb{L}(1)$. Each matrix $\mathbb{L}(j)$ can always be written as a product of an orthogonal matrix $\mathbb{Q}(j)$ and an upper right triangular matrix $\mathbb{R}(j)$, like a polar decomposition in radius and phase factor in the complex number space. The orthogonal matrix $\mathbb{Q}$ is like the phase factor and the $\mathbb{R}$ like the radius. $\mathbb{R}$ is the part of the matrix which becomes large and leads to the overall ill-conditioned behavior of Osedelec matrix. The idea is to write each $\mathbb{L}(\beth)$ as

$$\mathbb{L}(j) \cdot \mathbb{Q}(j-1) = \mathbb{Q}(j) \cdot \mathbb{R}(j) \tag{1.8}$$

where $\mathbb{Q}(O) = \mathbb{I}$, the identity matrix. This would give for the first part of the product

$$\begin{aligned} \mathbb{L}(1) &= \mathbb{Q}(1) \cdot \mathbb{R}(1) \\ \mathbb{L}(2) \cdot \mathbb{Q}(1) &= \mathbb{Q}(2) \cdot \mathbb{R}(2) \\ \mathbb{L}(2) &= \mathbb{Q}(2) \cdot \mathbb{R}(2) \cdot \mathbb{Q}(1)^T \\ \mathbb{L}(2) \cdot \mathbf{L}(1) &= \mathbb{Q}(2) \cdot \mathbb{R}(2) \cdot \mathbb{R}(1) \end{aligned} \tag{1.9}$$

and the next step would be

$$\mathbf{L}(3) \cdot \mathbf{L}(2) \cdot \mathbf{L}(1) = \mathbb{Q}(2) \cdot \mathbb{R}(3) \cdot \mathbb{R}(2) \cdot \mathbb{R}(1)$$

and for the full product

$$\mathbb{L}(t) \cdot \mathbb{L}(t-1), ..., \mathbb{L}(1) = \mathbb{Q}(t) \cdot \mathbb{R}(t) \cdot \mathbb{R}(t-1), ..., \mathbb{R}(1) \tag{1.10}$$

This is easily diagonalized, as the product of upper right triangular matrices is an upper right triangular matrix, and the eigenvalues of such a matrix are the numbers along the diagonal. So looking at the section 1.6, the Lyapunov exponent is read off the product of the upper triangular matrices as

$$\lambda_a = \lim_{t\to\infty} \frac{1}{t} \sum_{k=1}^{t} \log[R_{aa}(k)]. \tag{1.11}$$

This is the procedure for long times, but for the short ones is slightly different: when we make the recursive QR decomposition of $[\mathbb{L}^t(x)]^T \cdot \mathbb{L}^t(x)$ as

$$\mathbb{Q}_1(2t) \cdot \mathbb{R}_1(t) \cdot \mathbb{R}_1(t-1), ..., \cdot\mathbb{R}_1(1) = \mathbb{M}_1$$

if $\mathbb{Q}_1(t)$ were the identity matrix, the eigenvalues of the Osedelec matrix would all lie in the 'upper triangular' part of the QR decomposition. But for finite time $\mathbb{Q}_1(t)$ is not the identity in general. So we shuffle this $\mathbb{Q}$ factor over to the right of all the $\mathbb{R}$ matrices, and repeat our QR decomposition. For this we define a matrix $\mathbb{M}_2$ which has the same $\mathbb{R}$ factors as $\mathbb{M}_1$ but has the matrix $\mathbb{Q}_1(t)$ on the right:

$$\begin{aligned} \mathbb{M}_2 &= \mathbb{R}_1(t) \cdot \mathbb{R}_1(t-1), ..., \cdot\mathbb{R}_1(1) \cdot \mathbb{Q}_1(2t)) \\ \mathbb{M}_2 &= \mathbb{Q}_1^T(t) \cdot \mathbb{R}_1 \cdot \mathbb{Q}_1(t) \end{aligned} \tag{1.12}$$

and then we perform the recursive QR decomposition once again on $\mathbb{M}_2$:

$$\mathbb{M}_2 = \mathbb{Q}_2(t) \cdot \mathbb{R}_2(t) \cdot \mathbb{R}_2(t-1), ..., \cdot\mathbb{R}_2(1)$$

Since $\mathbb{M}_2 = \mathbb{Q}_1^T(t) \cdot \mathbb{M}_1 \cdot \mathbb{Q}_1(t)$, $\mathbb{M}_1$ and $\mathbb{M}_2$ have the same eigenvalues. Continue this sequence of operations creating $\mathbb{M}_3, \mathbb{M}_4, ..., \mathbb{M}_K$:

$$\mathbb{M}_K = \mathbb{Q}_K(t) \cdot \mathbb{R}_K(t) \cdot \mathbb{R}_K(t-1), ..., \cdot\mathbb{R}_K(1)$$

A theorem of numerical analysis states that as K increases, $\mathbb{Q}_K 2t)$ converges to the identity matrix. When $\mathbb{Q}_K(t)$ is the identity to desired accuracy, the matrix $\mathbb{M}_K$ is upper triangular to desired accuracy, and one can read off the Lyapunov exponents λ_a from the diagonal elements of the $\mathbb{R}_n(k)$'s:

$$\lambda_a = \frac{1}{t} \sum_{j=1} 2t \log[R_K(j)_a a]$$

since the eigenvalues of a product of upper triangular matrices are the product of the eigenvalues of the individual matrices.

1.3.5 Some remarks on the edge of chaos

His central role in the work make this concept deserving of a deeper insight.
The first contacts with the edge of chaos were at the convergence between biology and computing: in late 90s, Libermann and Conrad in their seminal works hint the existence of an optimal regime, or "sweet spot", where the living cells compute and transmit information following a trade-off between adaptability and efficiency.

But the first who properly mentioned the edge of chaos was Norman Packard, in his studies on cellular automata, a two-state units that evolve in time based on their neighbors through a set of rules. He observed two regimes in the cellulars' communication, one too rigid and not very sensitive and one in which any local perturbation affected all the cellular grid(more details in [24]). So he argued that the best computation rule lies near the transition between order and chaos. Few years later, Christopher Langton followed Packard's footsteps supporting his work and also going further: he defined a parameter λ that characterize the cellular automata rule space from most homogeneous to most heterogeneous rule tables; he was the first to wondered how his work was related to self-organized criticality.
Parallel to this first approach people started to investigate chaos and so clearer definition of edge of chaos came out. There are two main ways to determine what regime a system is in, one is the annealed approximation and the other is the Lyapunov exponent. In the first approach one creates two replicas of the same system, perturbing only one of them. The difference over time between the trajectories of the twp systems tells us what regime we are seeing: if this perturbation is dumped to zero the system is in his ordered phase; if the perturbation grows over time the regime is chaotic; if the difference remains constant we are on the edge of chaos.
For what regard Lyapunov exponent, as we saw in the previous sections, zero is the value that divides the two different regimes. We specify also that , in this section, we mean criticality as a dynamical concept; the critical point is the point that divide a predictable, periodic dynamics from a chaotic one.
In the followed years the concept of edge of chaos spread to various field, from biological, dynamical systems to sociological, economic systems, tying more and more with the ideas of self-criticality and phase transition ([24], [11]). At these days it remains something very present, especially in collective behavior (see bird flock, protein, brain cells) and computer science (see reservoir computer, neuromorphic computing); the edge of chaos remains a valid concept to investigate the surprisingly complicated phenomena occurring in his vicinity.

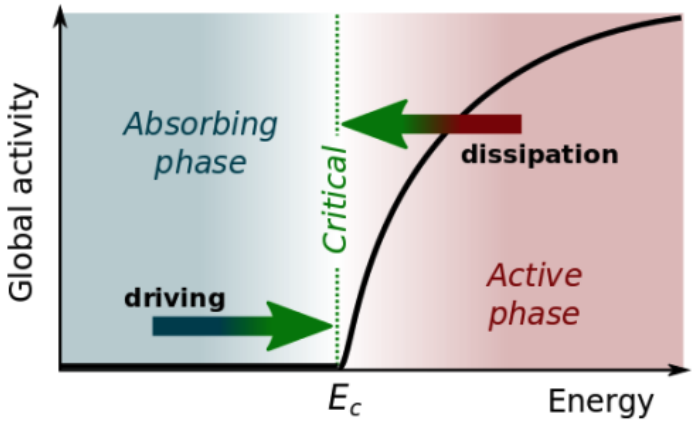


Figure 1.33. The self-organization-to-criticality (SOC) mechanism functions by creating a feedback loop between the dynamics of the system's activity and the control parameter (such as total accumulated energy, stress, or sand grains) across different timescales. Specifically, the control parameter becomes a dynamic variable that behaves differently depending on the system's state: rapid dissipation (negative force) prevails when the control parameter is in the active phase, while slow-driving dynamics (positive force) dominate in the absorbing or quiescent phase. This feedback process self-organizes the system toward the critical point of a second-order phase transition, provided the separation between the slow and fast timescales is infinitely large, and the dynamics remain conservative (Bonachela and Munoz, 2009; Vespignani et al., 1998, 2000; Zapperi et al., 1995).

Chapter 2

Underlying mechanism of chaos-driven neural network

After this dutiful premises we can resume the discussion begun in the first chapter introduction. We said we tried to actively bring the network on the edge of chaos during its training, in particular we made it through the implementation of Lyapunov exponent-based regularizer in the training loss. In fact, we just saw that Lyapunov exponent is an indicator of system's chaoticity, and so, if we consider the net as N-dimensional operator, it has a set of Lyapunov exponent which can be analytically computed and manipulated, making the network more converging or more chaotic and exploring.
The new loss function will be

$$Loss(y, \hat{y}) = MSE(y, \hat{y}) + \alpha \cdot |\lambda| \tag{2.1}$$

where α is a positive scalar and λ is the maximum Lyapunov exponent, the one that control the operator's behavior. In this way, the network will be forced to have a null λ value and so to be on the edge of chaos.
Though, during the course of our experiments, we asked ourselves if a more chaotic, and so explorative, behavior would give any improvements, and, if this is true, what is the limit. So we conducted the same experiments with another loss function:

$$Loss(y, \hat{y}) = MSE(y, \hat{y}) + \alpha \cdot \lambda \tag{2.2}$$

Taking away the module and using a negative value for α, during the training λ will be maximized without any bonds. Though, we don't expect that this last method will outperform the first overall: indeed, if being more chaotic could help in the transition phase, surely it don't help during standard learning, especially in the case of very simple deterministic dynamics, like a periodic one.
We then remark that the tendency of a neural network to have a negative Lyapunov exponent is something intrinsic in the training process: if we consider a simple training process with a set of inputs corresponding to precise outputs (like a classification problem), at the end of the training, a well trained network will associate a new data point near a certain training input to the output corresponding to the latter; in other words, points that are close in the inputs' space will be mapped to the same output. Moreover, we can show experimentally that also in our training process the

Lyapunov exponent of a neural network remains negative.
Now we are going to show the particulars of the experiment, passing from a toy model to chaotic systems.

2.1 Testing process

2.1.1 Experimenting on non-chaotic time series

In the first part of the experiment we considered a regime shift consisting in a sequence made up two sine waves, the easiest continuous nonlinear wave, with different frequencies and with or without a Gaussian noise, as one can see in Figure 2.2, 2.4, 2.3. In particular, we want verify if certain amount of noise can help Lyapunov network to better exploring the parameter space during transitions.
We trained two networks, identical in the structure and in the initial weights, to take a certain number of datapoints and predict the next one; one will be trained only with MSE loss and one will be trained with the new Lyapunov regularizer. In the following we refer to the first network as the vanilla network. For what regard the second one, the Lyapunov network, we applied **Theorem 1** in section 1.6 defining a loop in each training step, in which, taking a random initial condition, we see the evolution of this point and compute the linear stability matrix, or jacobian, in order to obtain the $\mathbb{P}$ matrix and λ.
Moreover, we mediated the results over a certain number of parallel trainings, in which the net with regularizer and the net without begin every time with the same initial weights.
In the second part of the experiment, beyond the frequency, we chanced also the wave form, taking a sine and a triangular wave, and than carrying out the same experiment.

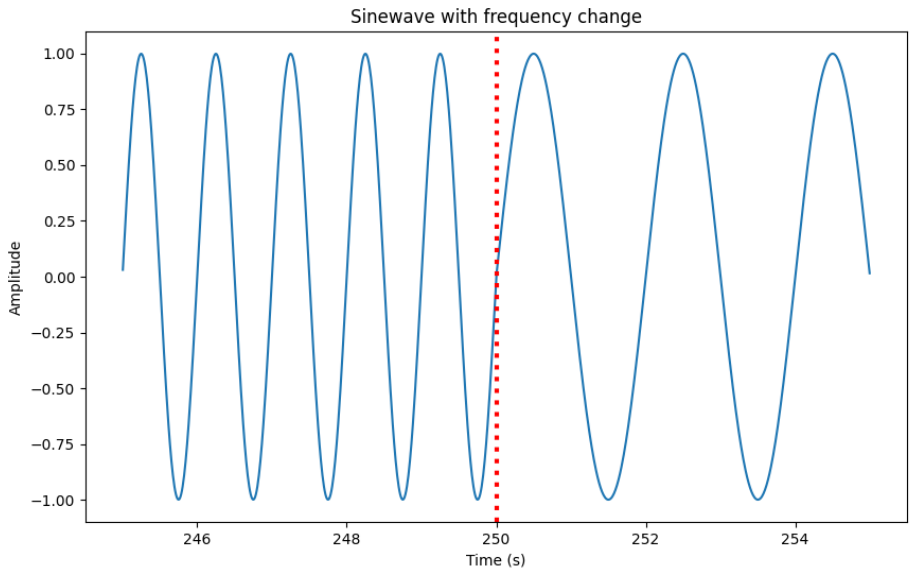


Figure 2.1. dataset example in the case of changing frequencies.

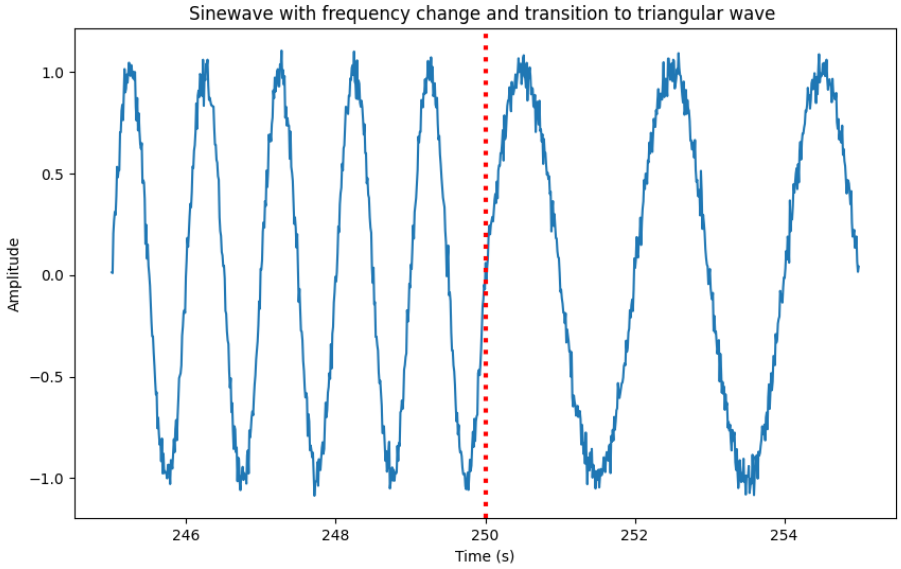


Figure 2.2. dataset example in the case of changing frequencies.

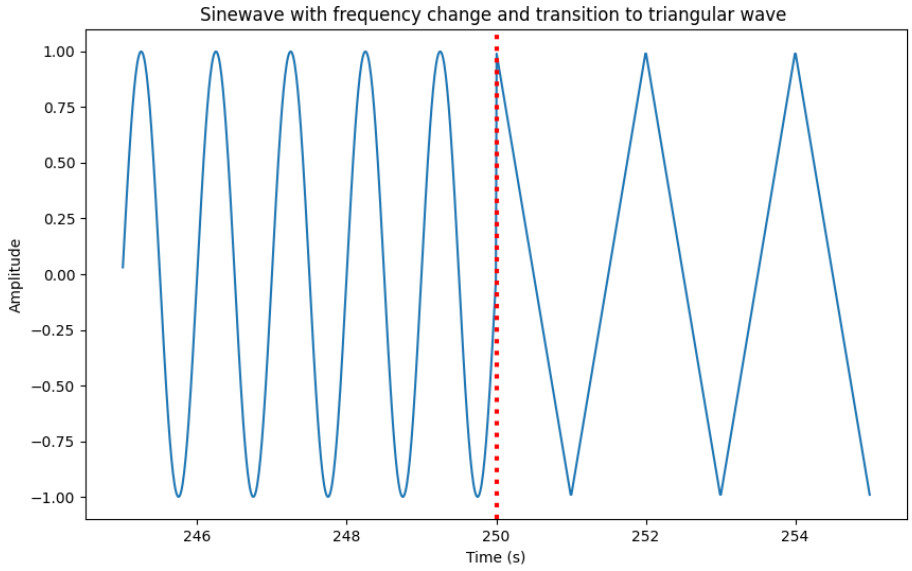


Figure 2.3. dataset example in the case of changing wave forms.

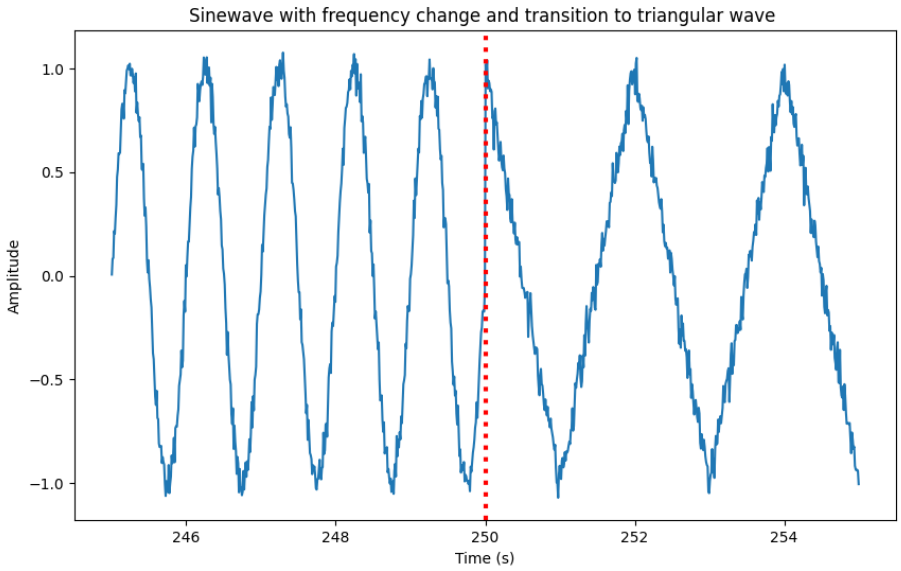


Figure 2.4. dataset example in the case of changing waveform with noise.

For the sake of clarity we specify the method implemented for the Lyapunov exponent computation: the recurrent prediction of the network doesn't make it possible too see the net as an $n \times n$ operator. So we can consider a function that associate 10 points to another 10 points translated by 1, that is

$$F(x_t, x_{t+1}, ..., x_{t+9}) = (x_{t+1}, ..., x_{t+10}) = (x_{t+1}, ..., f(x_t, x_{t+1}))$$

where the last, x_{t+10}, is neural network output. So the jacobian of this function is

$$\mathbb{L} = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ \frac{\partial f}{\partial x_t} & \frac{\partial f}{\partial x_{t+1}} & \frac{\partial f}{\partial x_{t+2}} & \dots & \frac{\partial f}{\partial x_{t+9}} \end{pmatrix}$$

Considering this as the evolution map in our training process, we were able to compute the Lyapunov exponent.

2.1.2 Experimenting on chaotic time series

After these simple, preliminary case studies, we investigated much more difficult time series, the dynamics of Lorenz system, the most known and famous dynamical system that exhibit chaos. It will allow us to test our regularizer both on regime shift and on predicting capacity over an unpredictable phenomenon like a chaotic system.
With the same setting, we take two sequences generated changing the control parameter of the system, in order to obtain different regime shift, i.e. ordered-chaotic

or chaotic-chaotic, as shown in Figure 2.5 ans Figure 2.6. So again we train two identical feed forward neural networks in the two different modalities described above. Given the chaotic nature of the system, in this case the presence of the noise would be unnecessary and detrimental.
Another important difference is in the computation of Lyapunov exponent: in this case we use the QR decomposition in place of **Theorem 1**, described in section 1.6, more stable and effective.

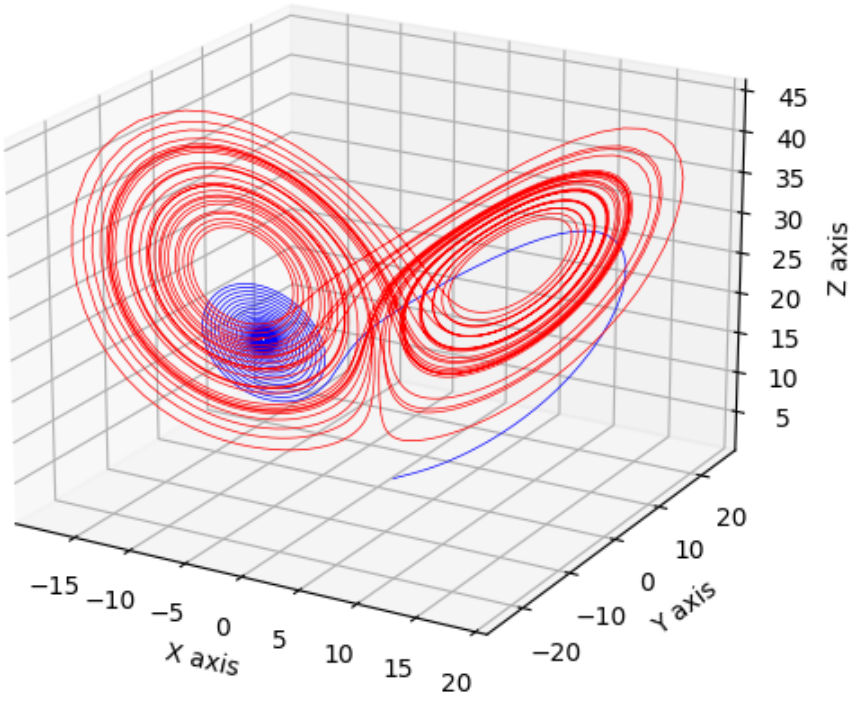


Figure 2.5. first 50000 steps of the training sequence before and after the parameters changing in the case of nonchaotic-chaotic transition.

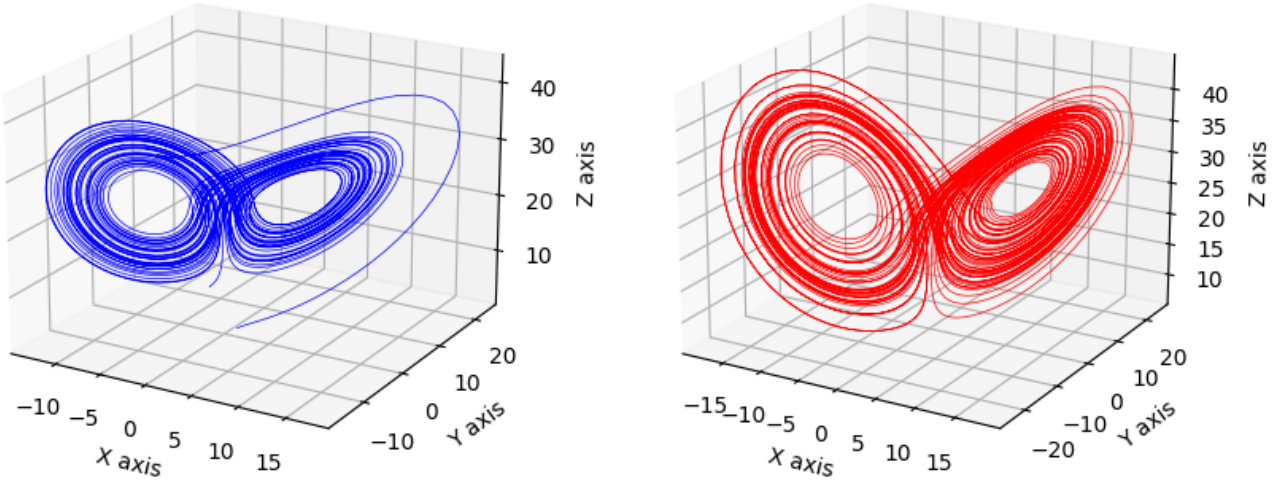


Figure 2.6. first 50000 steps of the training sequence before and after the parameters changing in the case of chaotic-chaotic transition.

2.2 Evaluation method

When one is dealing with different neural networks, often confronting their loss score is the best choice for evaluating performances. So, in order to determine the positive or negative outcome of our experiment, we chose the training loss. In particular, we looked at the ratio between the two losses in, and immediately after, the transition point, making evident how much the regularized network overcome or not the vanilla performances.
Moreover, we took care to verify that the network effectively learn the task, as reaching better performance with respect the vanilla network without the correct learning would be only a partial success.
Maybe one can think to use convergence time of the loss to compare the networks behavior, but it's not properly what we are looking for: what really we want is to verify that this regularizer make the network more prepared to sudden events, so the ratio in the transition point is the best candidate.
This choice is also quantitatively supported by [21] in which the authors showed that the neural gradients are lognormal distributed, that say the gradient logarithm values are normally distributed, as opposed to the gradients themselves. In fact, if we want to evaluate the distance between two quantities lognormal distributed, we have to take its ratio in order to know the difference between exponent's argument.

Chapter 3

Results

In this chapter we present the main results of the experiments described above, starting from the simple sine and triangular waves up to the Lorenz equations.

3.1 Sine and triangular waves

First we present the training results obtained from the sine-to-sine transition, with the Lyapunov exponent which tend to zero using 2.1. The training had the following characteristics and was averaged over 20 simulations:

- first sine wave have a frequency of 1 Hz, the second one a frequency of 0.5 Hz, as shown in Figure 2.3, 2.4, 2.2;
- the network take a sequence of 10 points and give 1 point prediction, so it has 10 neurons in input, 10 neurons for the hidden layers, 1 neuron in output;
- we use Adam as optimizer and a learning rate of 0.01;
- we use a 20 steps loop for the exponent computation;
- 20 batch points for the MSE loss computation;
- tanh activations between the 2 hidden layers.

All this parameters are a trade off between accuracy and computation capabilities, being the training process with Lyapunov regularizer very time consuming with respect the vanilla training. Obviously the parameter space it's all to be explored but we don't expect a radical qualitative change in the results we obtained.
In Figure 3.1, 3.2 and 3.3 one can observe the first result from the experiment, where is immediately evident the difference between the two losses. In particular, in Figure 3.2, one can see that, after the first phase in which almost both converge at the loss minimum, the spike of the vanilla loss is higher than the other one. Moreover, the Lyapunov learning loss converges slightly faster toward its minimum, but unfortunately this minimum is grater than the vanilla minimum, as one can see in Figure 3.3: this is in part expected due to the very definition of Lyapunov exponent, given that something with a greater exponent will have much more difficult

to converge. This fact also suggest, at least in this cases, an adaptive use of this regularizer, in order to turn it on only when it's necessary or under specific conditions.

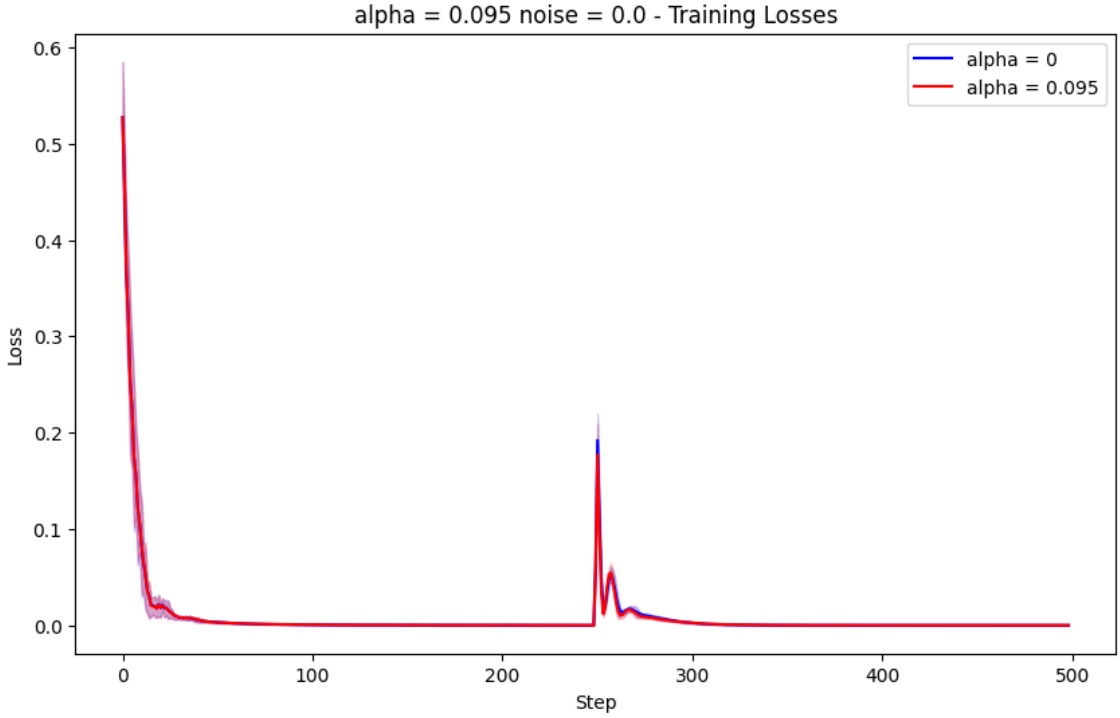


Figure 3.1. training loss from edge of chaos training obtained from averaging on 20 simulations, with an α value of 0.095 and without noise; the shaded areas represent the third and first quartile of the two quantities.

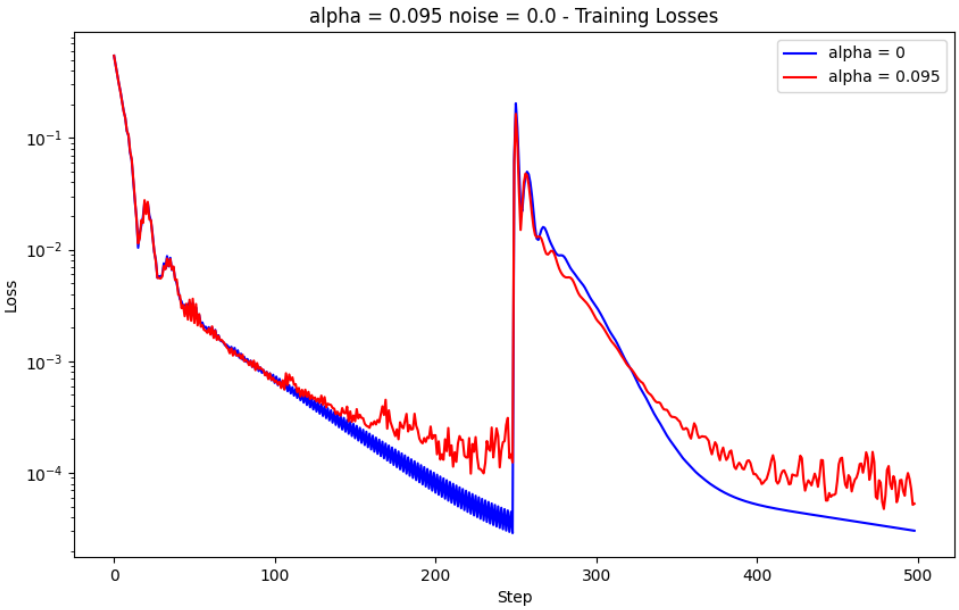


Figure 3.3. training loss from edge of chaos training in logarit hmic scale,obtained from averaging over 20 simulations, with an α value of 0.095 and without noise.

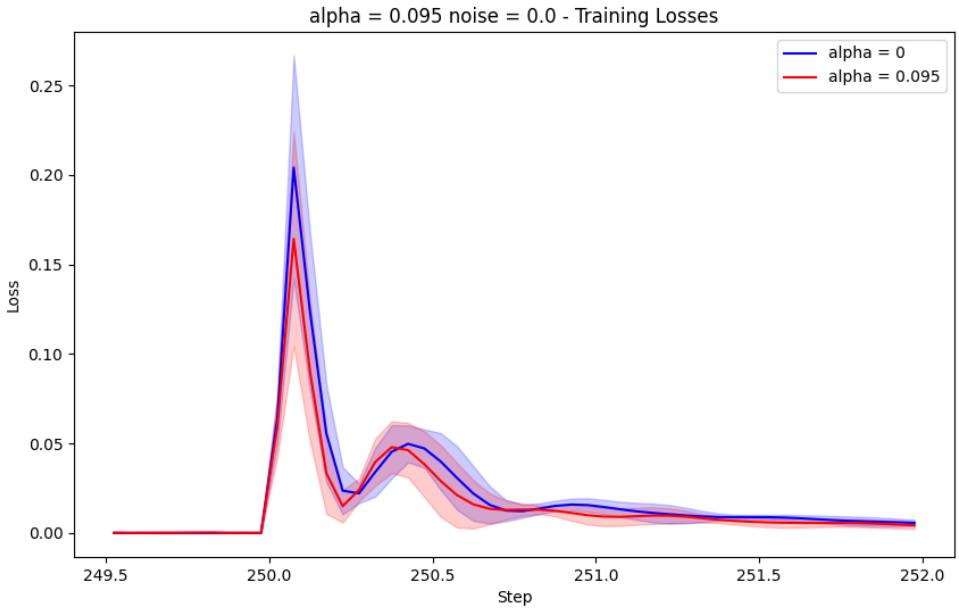


Figure 3.2. training loss from edge of chaos training focusing on the transition zone,obtained from averaging over 20 simulations, with an α value of 0.095 and without noise; the shaded areas represent the third and first quartile of the two quantities.

Figure 3.5, beyond confirming the intuition expressed in the last chapter, clarify

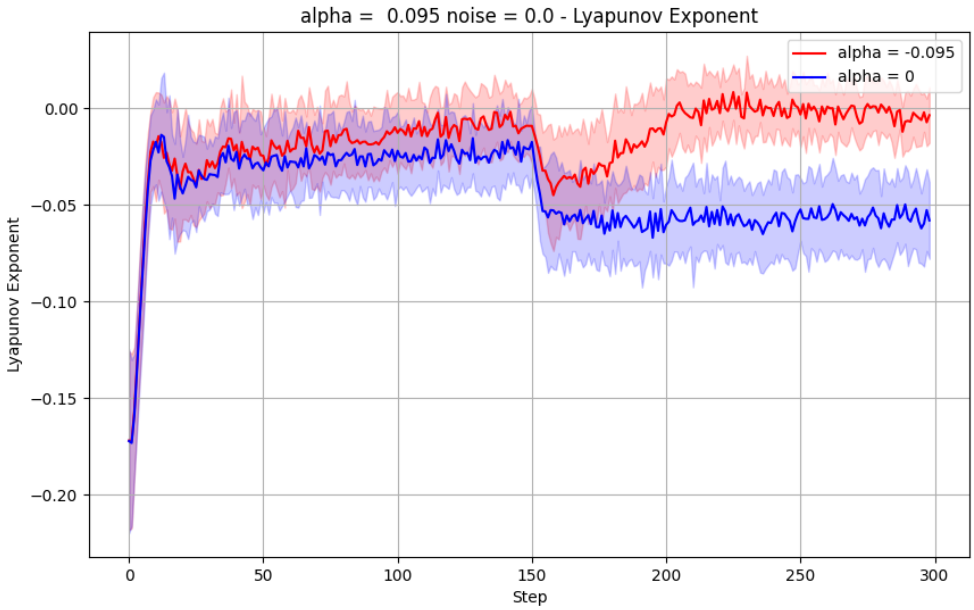


Figure 3.4. Lyapunov's exponents of the vanilla and chaos driven network over training steps in the case of edge of chaos training; the shaded areas represent the standard deviation of the two quantities.

what just we said above: one can appreciate the advantage of the neural network with the Lyapunov regularizer during the transition phase with respect the vanilla network. So a certain amount of chaos can help the network to be more resilient in the more difficult training phase, or, in other words, it help the network to explore the parameters space in order to find a configuration which better satisfy the loss constrains.
But at the same time, having a Lyapunov exponent greater than 0 during a phase free of regime shift is quite deleterious as we anticipated, see Figure 3.6.

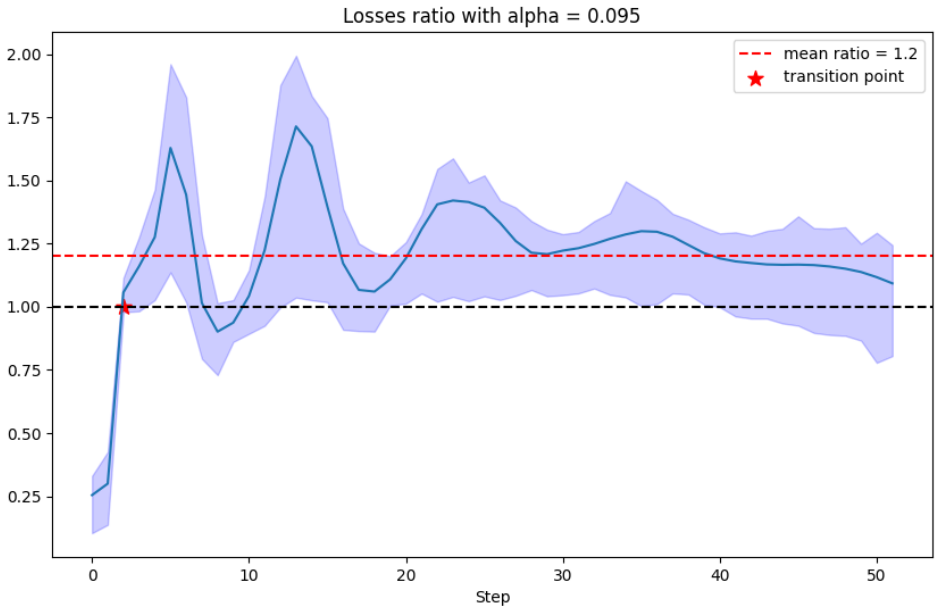


Figure 3.5. ratio between the averaged losses of the vanilla and chaos-driven experiment in the transition phase, in the case of edge of chaos network; the shaded areas represent the third and first quartile.

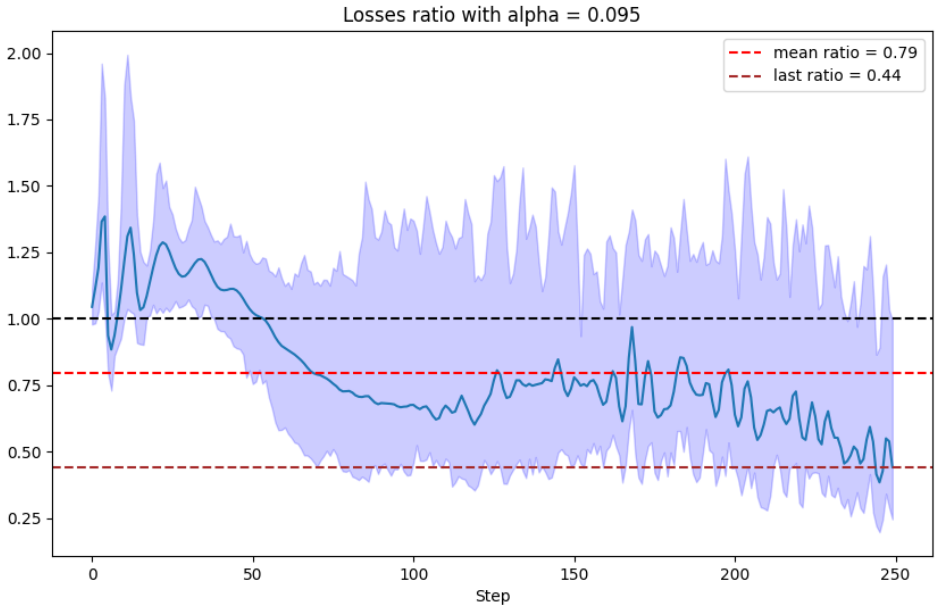


Figure 3.6. ratio between the averaged losses of the vanilla and chaos-driven experiment after the transition phase, in the case of edge of chaos network; the shaded areas represent the third and first quartile.

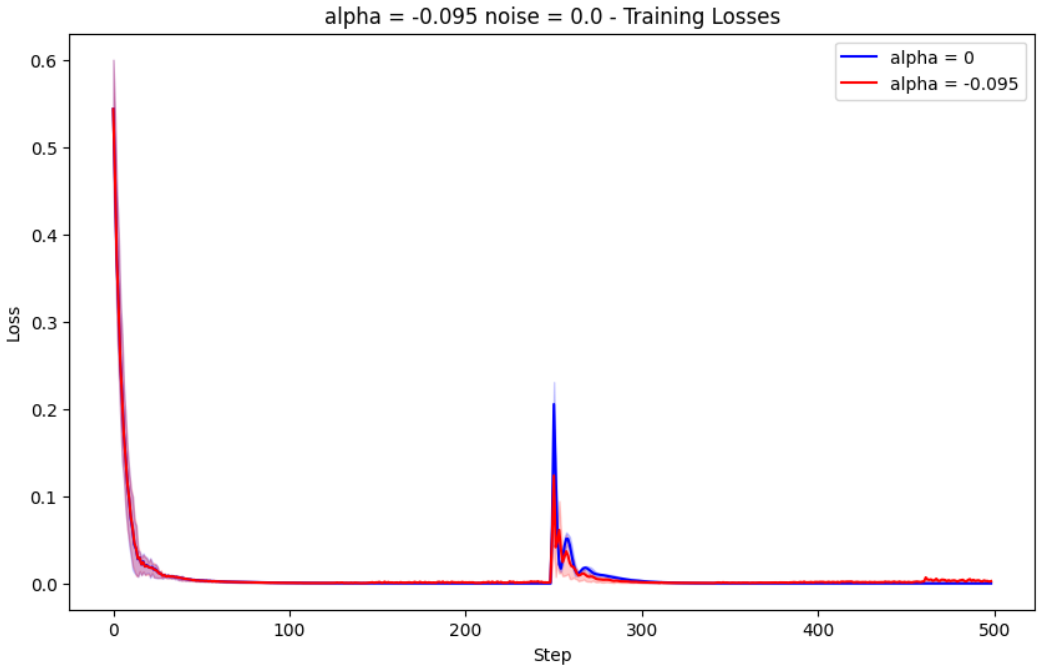


Figure 3.7. training loss of the two network obtained from averaging over 20 simulations, with an α value of -0.095 and without noise; the shaded areas represent the third and first quartile of the two quantities.

Now we present the results from the maximization of the Lyapunov exponent obtained applying 2.2.
In Figure 3.7, 3.8, one can observe that the results are qualitatively similar from the previous ones. However, in this case the effects of the regularizer are magnified, as one can see from Figure 3.9, 3.11, 3.12: in the transition zone the regularized network reaches a peak of more than 400% better performances with respect the vanilla network, maintaining this advantage in the next 73 training steps. But to this enhancing correspond also a grater downside: as one can see from Figure 3.12, 3.9, although reached faster, the minimum of the regularized network is an order of magnitude greater than the one of the vanilla network. Anyway, this result are not surprising, given that more chaos means more adaptability but also more fluctuations and instability, as testified also by the great variance of this processes.

In the end, we can't affirm that one procedure is better than another, instead we found further confirmations that a training with alternating phase of more and less chaotic behavior would be a better solution.

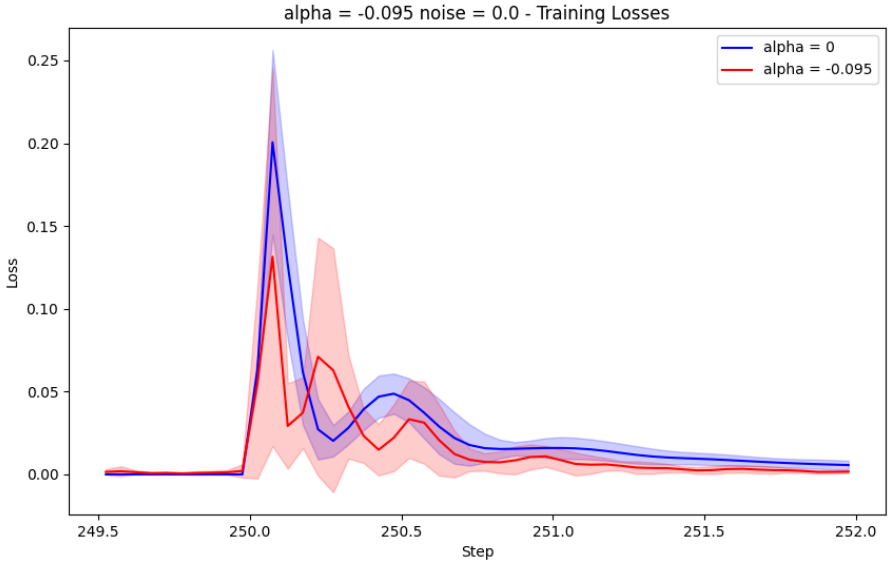


Figure 3.8. training loss of the two networks focusing on the transition zone,obtained from averaging over 20 simulations, with an α value of -0.095 and without noise; the shaded areas represent the third and first quartile of the two quantities.

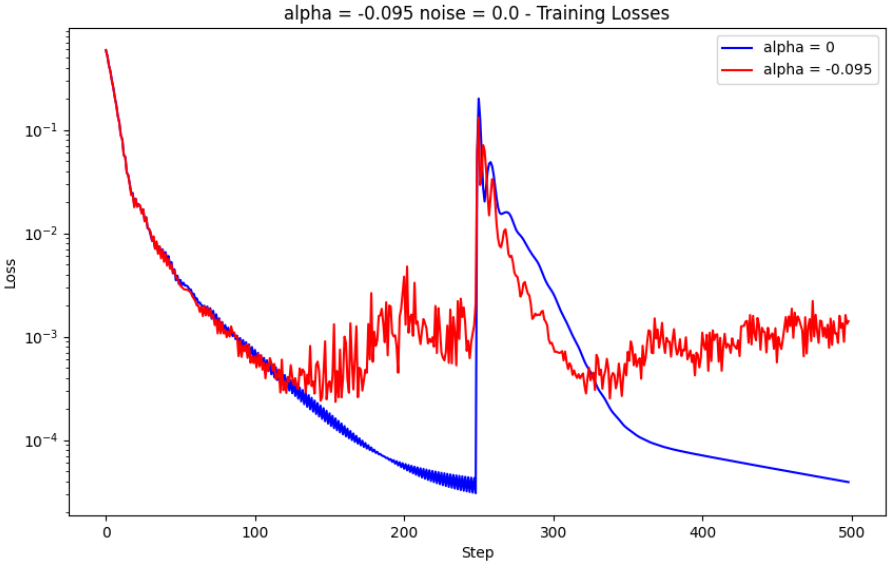


Figure 3.9. training loss from the maximization training in logarithm scale, with an α value of -0.095 and without noise.

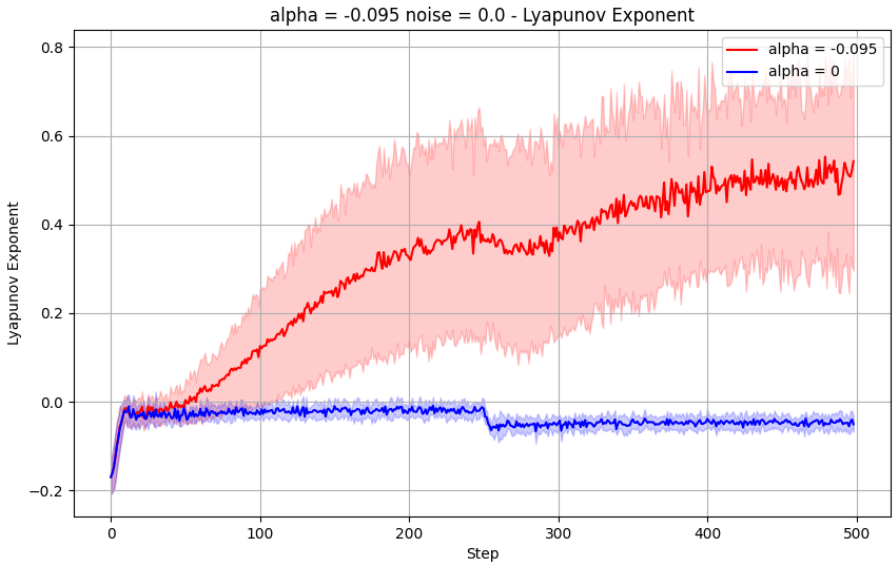


Figure 3.10. Lyapunov's exponents of the vanilla and non vanilla network over training steps in the case of exponent maximization training; the shaded areas represent the standard deviation of the two quantities.

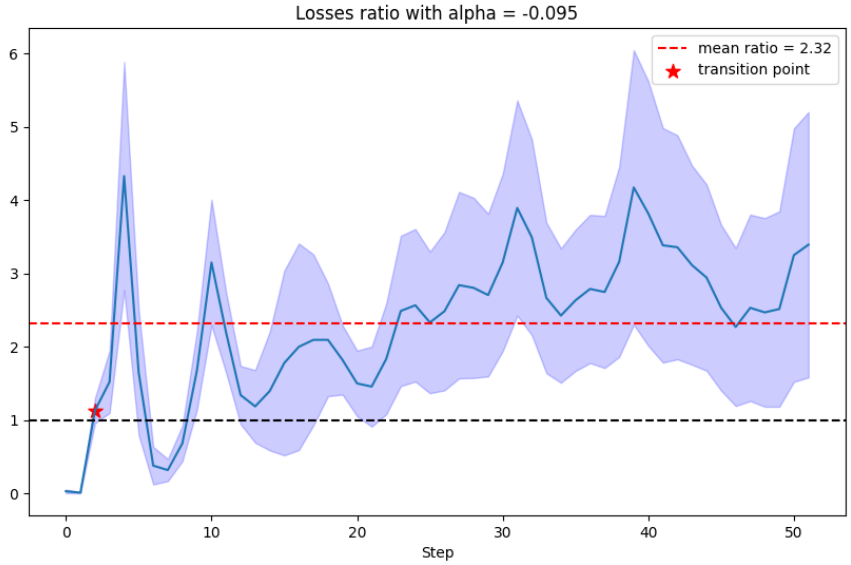


Figure 3.11. ratio between the averaged losses of the vanilla and non vanilla experiment in the transition phase in the case of exponent maximization training; the shaded areas represent the standard deviation.

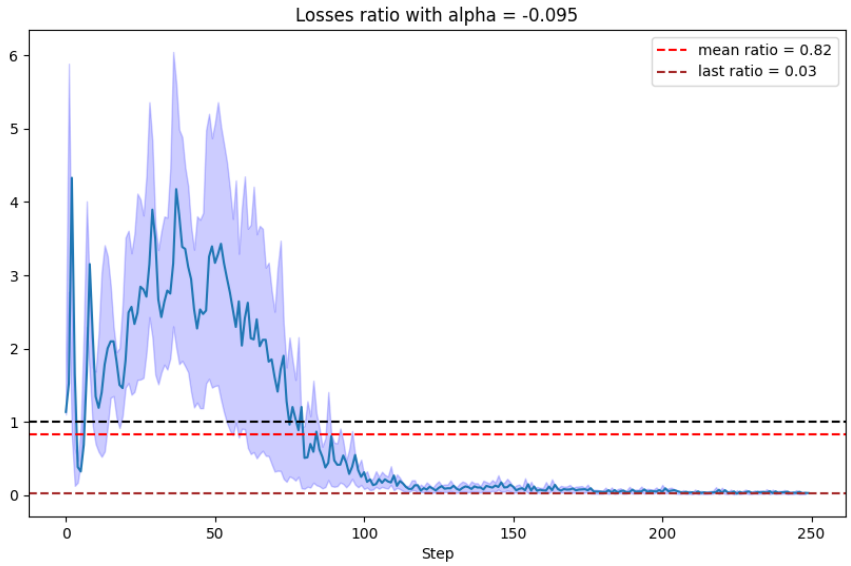


Figure 3.12. ratio between the averaged losses of the vanilla and non vanilla experiment after the transition phase in the case of exponent maximization training; the shaded areas represent the standard deviation.

It's interesting to note also the regularizer effects on the network's exponent and the behavior of a non-constrained network: in one case the exponent tend to zero(Figure 3.4) or keep increasing(Figure 3.10), as we want in the different training processes, and in the other one it stabilizes on a certain negative value for not clear reason yet. But in both cases in the point corresponding to the transition there is a macroscopic fluctuation toward a less 'exploring' behavior. We note also note that the measures we made, in particular for the regularized network, are affected by a large variance, attributable to the chaotic nature of this phenomena.

Obviously it depends on the task the network have to do, but in our case, although the big difference in the final phase, the performances of the two nets are nearly the same: we evaluated in Figure 3.13 this performances taking the prediction from the networks in the same way they were trained, so ten sliding points to predict the next one.

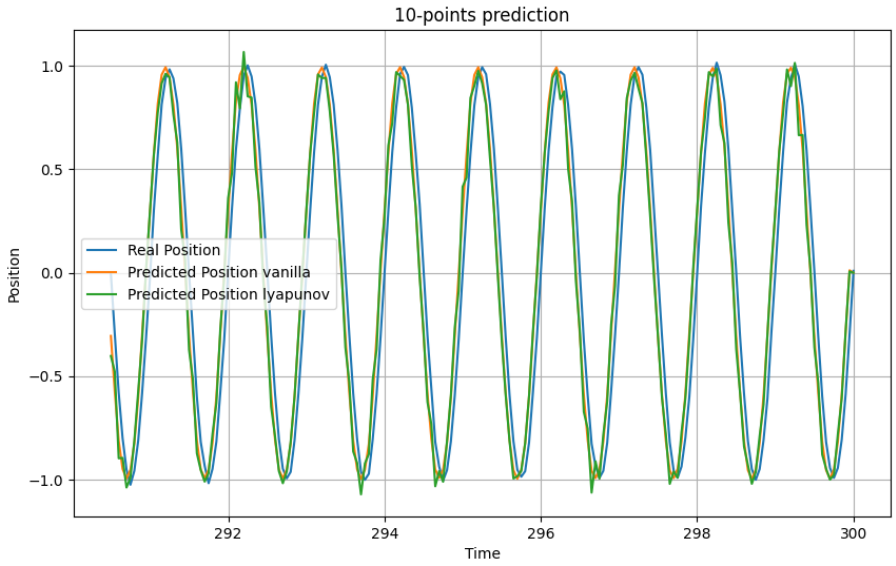


Figure 3.13. predictions of the vanilla network and Lyapunov network taking 10 points at a time to give the next one after the training, taking the weights associated to the best performance.

Since we observe an effective improvements in exploration capacity in the case of the maximization exponent, we tried to push forward this aspect through noise, in order to verify if random fluctuations amplified the observed effects.
So we run simulations for different values of noise amplitude and α, in order the find the best values. The y-values plotted in 3.15 and 3.16 are the mean ratios obtained from the simulations with that particular values of α and noise.
From these simulation emerge that the regularizer is robust with respect the noise, as the effect persist as noise increases, but we also observe that the benefit from the Lyapunov regularizer decreases, contrary to what we thought. So noise, differently from chaos, don't give any improvements in term of network elasticity, but only it makes the task more difficult for all the two networks. At the end of the day we found that, for every noise amplitude, the best ratio values are always in the range of α [0.05, 0.1], probably because the noise we introduce is only additive and so don't change the qualitative characteristic of the studied phenomena, until this noise is too large to suppress the signal obviously.

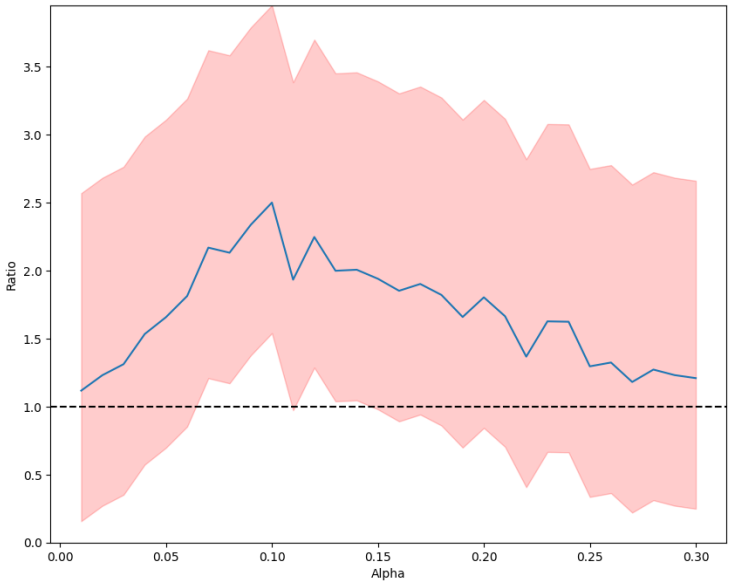


Figure 3.14. The ratio between the losses of vanilla and chaotic-driven network as function of the value of α in the case of two sine waves with different frequencies and in case of zero amplitude noise, obtained from 20 simulations; the shaded area in red is the standard deviation.

Looking at 3.14 we can recover the adjacent possible concept: here the value of α is a measure of how far the exploration of the net can go in his research of some novelties capable of decrease the loss function. So, if this value is too large the network cross the frontier of the adjacent possible going in the far possible space, and reaching phase space regions too far from those where the net learned, getting worse performance. To be more explicit, the far possible is technically all the elements far more than one steps in a graph, and if one explore too far can reach elements that seems uncorrelated to what you actually know.

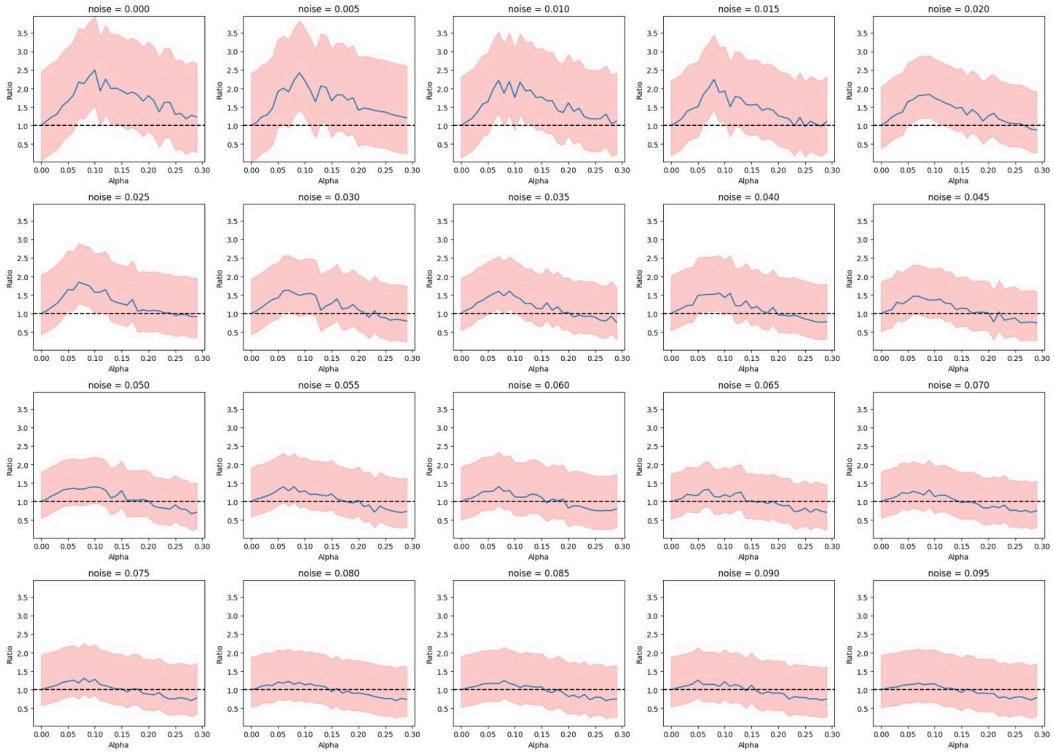


Figure 3.15. The ratio between the losses of vanilla and chaotic-driven network as function of the value of α in the case of two sine waves with different frequencies, for different value of noise amplitude; the shaded area in red is the standard deviation.

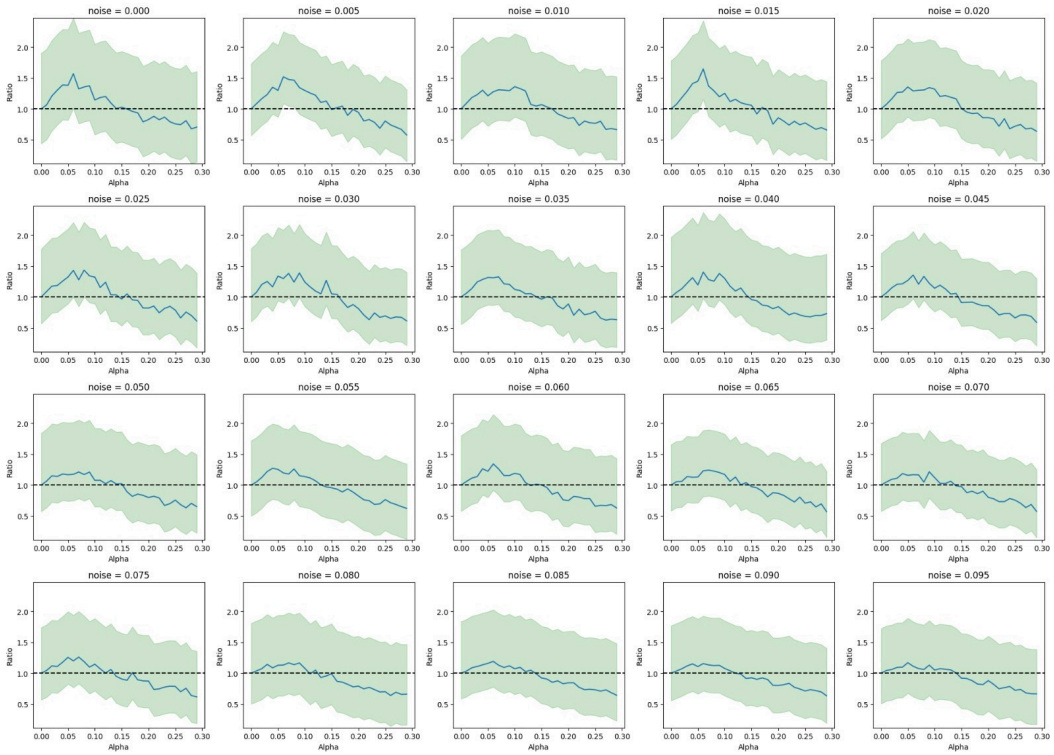


Figure 3.16. The ratio between the losses of vanilla and non vanilla networks as function of the value of α in the case of sine and triangular waves with different frequencies, for different value of noise amplitude; the green shaded area is the standard deviation.

3.2 Lorenz equations with changing parameters

Now we present the results on Lorenz system. We averaged over 15 simulations with the same methodologies as before and with these parameters:

- for the first case, we have two time series obtained from two Lorenz systems, one have as parameters $\sigma = 10$, $b = 8/3$, $r = 20$, non-chaotic, and one with $\sigma = 10$, $b = 8/3$, $r = 28$, chaotic, as shown in Figure 2.5 ans 2.6;
- for the second case, we have two time series obtained from two Lorenz systems, one have as parameters $\sigma = 10$, $b = 4/3$, $r = 25$, chaotic, and one with $\sigma = 10$, $b = 8/3$, $r = 28$, chaotic
- the network take one three-dimensional point and give one point prediction, so it has 3 neurons in input, 50 for the hidden layers (4 layers in particular), and 3 in output;
- we use Adam as optimizer and a learning rate of 0.001;
- we use a 10 steps loop for the exponent computation and the QR decomposition;
- tanh activations between hidden layers;

- 50 batch points for the MSE loss computation.
- an α value of 0.095, suggested by preliminary simulations made around this value.

Again, these parameters are a trade off between accuracy and computing capacity, being these simulations even more time consuming compared to the precedents. Figure 3.17 is very similar to the results we showed above: one initial phase with the 'easy' task in which both network converge to the same minimum, then the transition spike and the slow settle down toward the new minima. But in this case it's interesting to note that, with respect the other section, the good result of the experiment are far more evident.

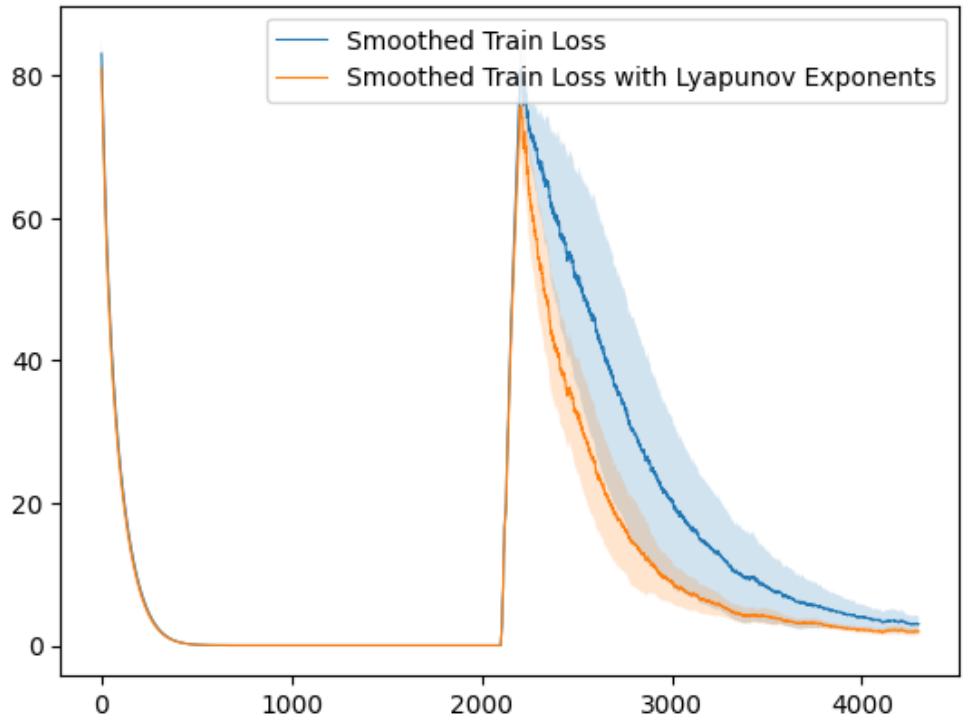


Figure 3.17. training loss means of the two network obtained over 15 simulations, vanilla and with Lyapunov regularizer, in the case of nonchaotic-chaotic transition; the shaded area represent the third and first quartile; in this case, in order to visualize the results with less fluctuations we did moving average smoothing.

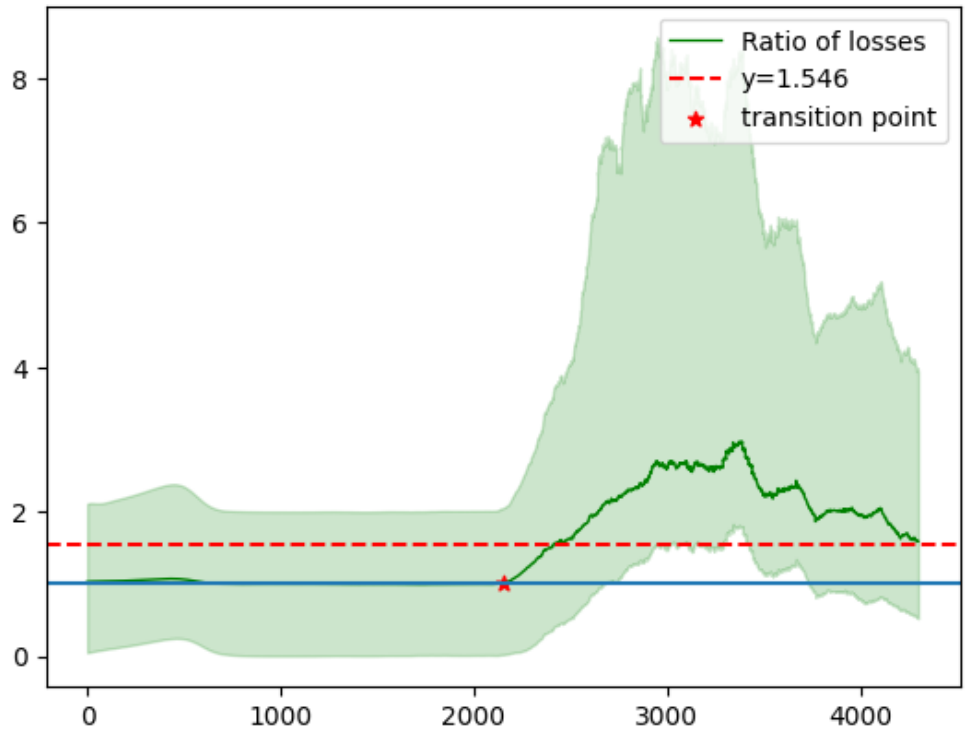


Figure 3.18. ratio between Lyapunov training loss and vanilla training loss, in the case of order-chaos transition; the shaded area represent the third and first quartile, the red dotted line is the mean ratio over the total training; in this case, in order to visualize the results with less fluctuations we did moving average smoothing.

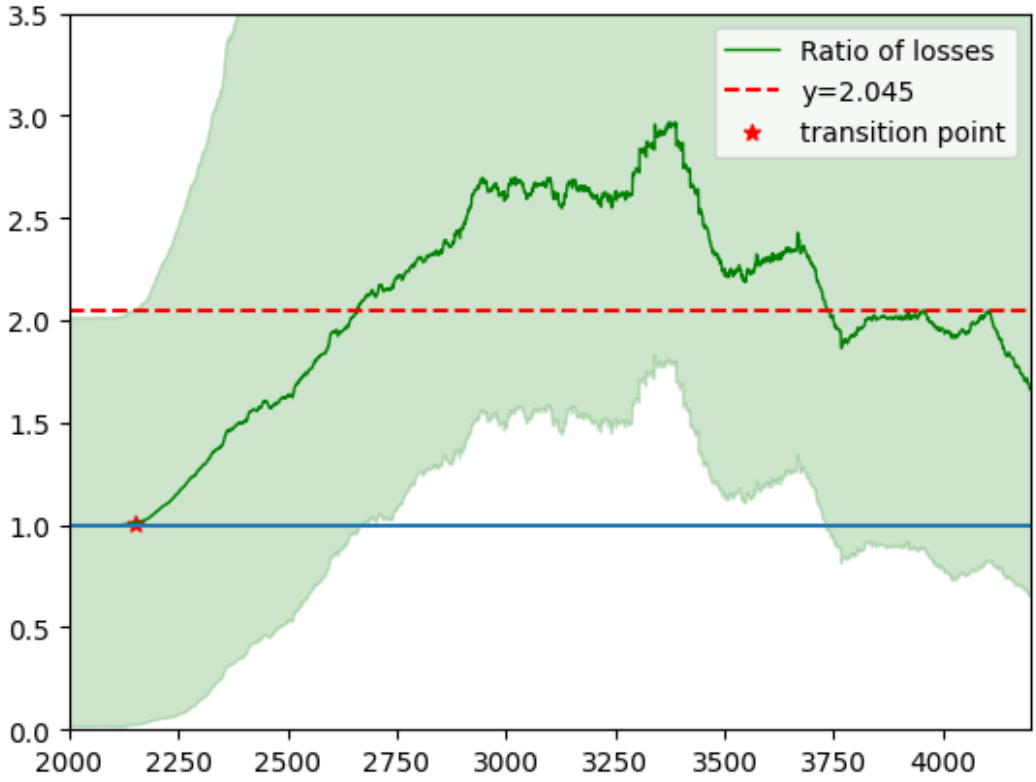


Figure 3.19. a zoomed view of the ratio between Lyapunov training loss and vanilla training loss, in the case of order-chaos transition; the shaded area represent the third and first quartile, the red dotted line is the mean ratio over the training phase after the transition point; in this case, in order to visualize the results with less fluctuations we did moving average smoothing.

Even just from Figure 3.17 it's easy to observe the better performance of the Lyapunov-regularized network with respect the vanilla one: immediately after the transition point and all the way after, our network is has the better loss score. We also note that, contrary to the precedent section, the regularized network has a way less variance compared to the vanilla one, as if the Lyapunov network were more prepared to the chaos it will see in second phase. Precisely the goal we wanted to achieve.
From Figure 3.18 and Figure 3.19, although the big variance that could partially invalidate the results, the observation above it's clear, our network achieves almost 300% better performance. In particular in Figure 3.19, where is evident the effect of the regime shift, is appreciable the crucial role of the regularizer. The positive fact about the variance is that in the transition zone is where it's less compared to the zone ahead.

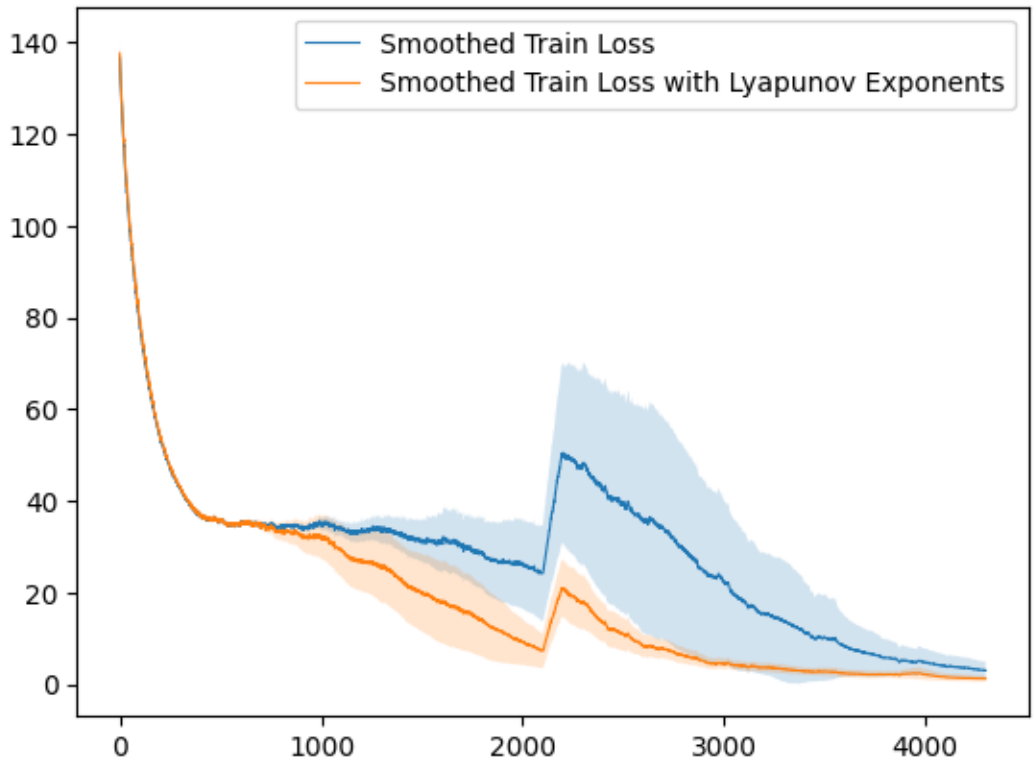


Figure 3.20. training losses of the two network, vanilla and with Lyapunov third and first regularizer, in the case of chaotic-chaotic transition, obtained from averaging over 15 simulations; the shaded area represent the standard deviation and in this case, in order to visualize the results with less fluctuations we do moving average smoothing.

In the case of chaotic-chaotic transition it seems that the regularizer effect is even more evident, also before the transition point, as one can see in figure 3.20, 3.21, 3.22. This is the confirmation that, beyond where a transition occurs, when the task is particular difficult, like in the chaotic regime, the help of this regularizer is crucial to obtain better performances.
Figure 3.20 is quite revealing: at beginning of the training there is a moment of hesitation for both the networks in which it seems that they settle down on a loss minimum, but after some steps the regularized network come out of this spurious minimum and keep searching, despite the the vanilla one that slowly realize his error. Then the Lyapunov network maintain his gain on the spike and on all the way after. Also the variance much less with respect the vanilla case as in precedence.
Moreover, the prediction point by point obtained from the regularized network is quite close to the original trajectory, as one can see in Figure 3.23.

Given all this results, we can conclude this chapter by safely affirming that the initial intuition, though not support by rigorous demonstrations, is experimentally correct: it's possible magnify neural network resiliency and performance by widen its exploring capacity through chaos.

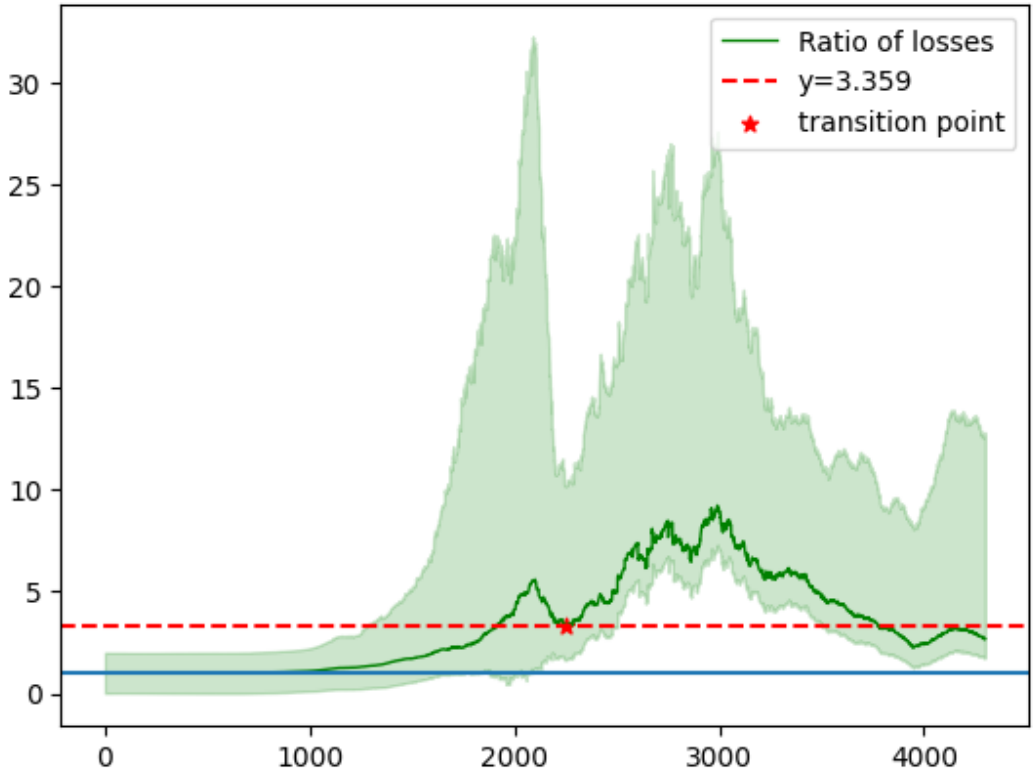


Figure 3.21. ratio between Lyapunov training loss and vanilla training loss, in the case of chaos-chaos transition; the shaded area represent the third and first quartile, the red dotted line is the mean ratio over the total training; in this case, in order to visualize the results with less fluctuations we did moving average smoothing.

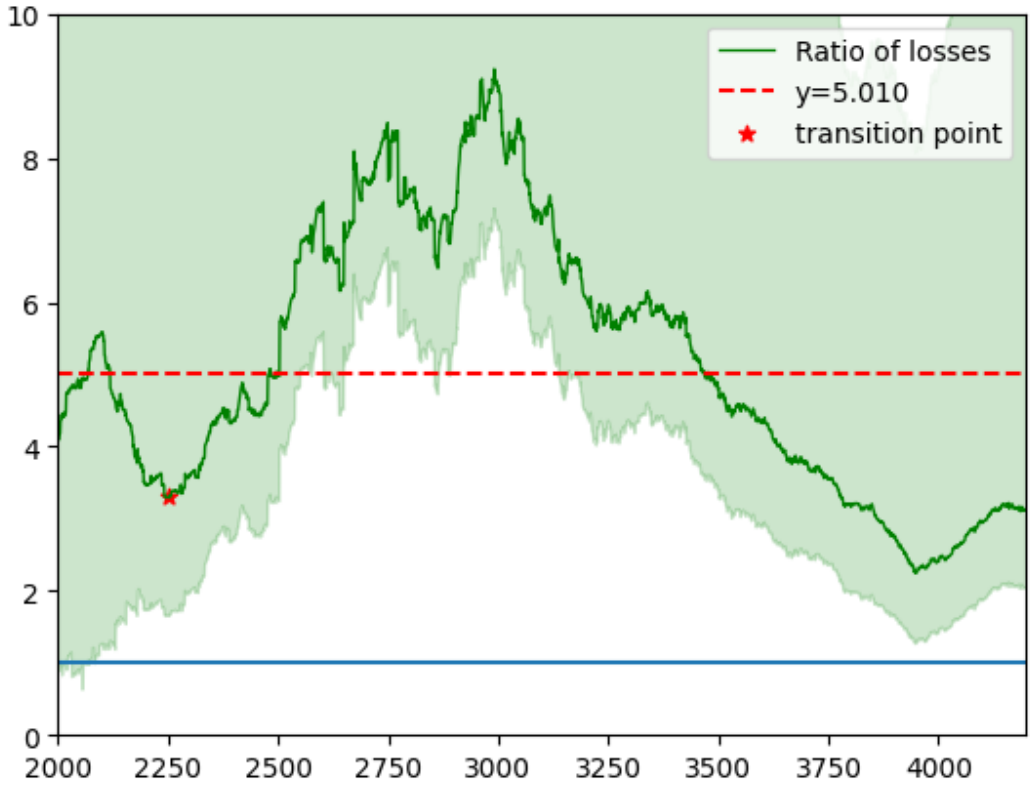


Figure 3.22. a zoomed view of the ratio between Lyapunov training loss and vanilla training loss, in the case of chaos-chaos transition; the shaded area represent the third and first quartile, the red dotted line is the mean ratio over the training phase after the transition point; in this case, in order to visualize the results with less fluctuations we did moving average smoothing.

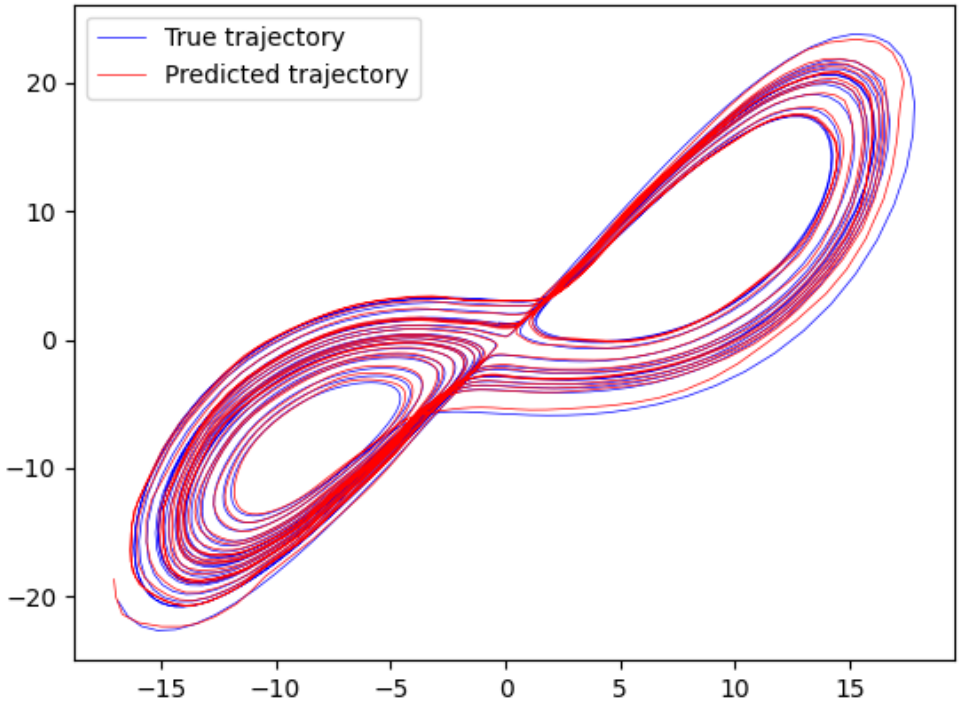


Figure 3.23. comparison between the point by point prediction of the regularized neural network and the true trajectory.

Chapter 4

Conclusions

We began our discussion by talking about how our lives are in contact with phenomena characterized by sudden changes in behavior, which seem to be the main way through which evolution progresses over time. Given the ubiquity of this phenomenon, developing a method or algorithm capable of quickly adapting to these changes would not only be helpful in technical and scientific fields but could also improve the quality of life for people in certain areas, especially when considering sudden illnesses or, now more than ever, climate change.
In light of these considerations, we sought to address this problem starting from both simple and complex dynamic systems, in order to have a testing ground for our idea in view of future applications to real phenomena. Aware that in the literature the edge of chaos is often associated with improved capabilities, both in biological systems and in neural networks, we employed this concept in a new way, developing a regularizer capable of bringing a network into this condition. A network with these capabilities could not only be used in situations like regime shifts but could also help in unresolved problems of neural networks, such as catastrophic forgetting. Furthermore, the exploration capabilities of the adjacent possible could make this method appealing for innovation research in various fields.

Specifically, we tested our idea first on simple periodic waves, changing the type of wave and frequency, then on a dynamic system, the Lorenz system, which can exhibit purely dissipative or chaotic behaviors. The two experiments were conducted in two different modes: the performance of a network trained without a regularizer was first compared with those obtained at the edge of chaos, and then with those obtained by trying to maximize the Lyapunov exponent of the network, pushing it beyond the edge of chaos. What we observed is that by modulating the strength of the regularizer at certain values, it allows for achieving performance improvements by various multiples, reaching a mean gain of 336% in the case of transitions between two chaotic regimes. However, the most positive effects are observable only regarding chaotic systems: indeed, while with the latter it may be useful to increase the exploratory capabilities of the network, this behavior will certainly be detrimental in the learning of an exclusively deterministic system. This is observed in the final training phases of the first experiment, where the loss of the regularized network stabilizes at least one order of magnitude above that of the

non-regularized network. This effect is amplified by the second training method, which maximizes the exponent: performance improves significantly, more than 232% higher than the normal case in average, but deteriorates equally by the end of training.

It is interesting to note that, by introducing additive Gaussian noise, the effect of the regularizer is robust to it, and furthermore, this type of noise does not seem to benefit the elasticity of the network but rather worsens the performance for both differently trained networks.
The results are encouraging but lack a solid theoretical foundation that explains these effects; one possibility could be to place this approach in the context of dynamic systems, treating the neural network as an n-dimensional map that induces a certain dynamics, as we have begun to do in this experiment. Moreover, the behavior of the Lyapunov exponents of a simple non-regularized neural network is not well understood, and shedding light on this phenomenon could aid in the study of neural networks themselves.
It could be useful to develop an algorithm capable of modulating the regularizer's strength in an efficient way, in order to favor a chaotic behavior only when necessary, making this chaos-driven neural network a possible solution for the continual learning problem.
Besides this possible implementation, would be interesting to combine this approach to the recent development to solve the catastrophic forgetting, and so the chaos driven approach to the neuronal plasticity-inspired learning rate algorithm; not only could be an improvement for the neural networks but also for our knowledge of human brain.
We highlight also the important achievement of exploring the adjacent possible in the case of continuous systems, never implemented before and more suitable when dealing with natural phenomena. Of course, the natural evolution of such work is the application to the aforementioned regime shift phenomena, using data collected directly from the world around us, in the hope of improving artificial intelligence and our response to often uncontrollable phenomena.

Bibliography

[1] Lin Zhang, Ling Feng, Kan Chen and Choy Heng Lai, "Edge of chaos as a guiding principle for modern neural network training", July 2021, https://doi.org/10.48550/arXiv.2107.09437

[2] Ryan Vogt, Maximilian Puelma Touzel, Eli Shlizerman, Guillaume Lajoie, "On Lyapunov Exponents for RNNs: Understanding Information Propagation Using Dynamical Systems Tools", June 2020, https://doi.org/10.48550/arXiv.2006.14123

[3] Anurag Dutta, John Harshith, A. Ramamoorthy, K. Lakshmanan, "Attractor Inspired Deep Learning for Modelling Chaotic Systems", November 2023, https://doi.org/10.1007/s44230-023-00045-z

[4] L. Storm, H. Linander, J. Bec, K. Gustavsson and B. Mehlig, "Finite-time Lyapunov exponents of deep neural networks", June 2023, https://doi.org/10.48550/arXiv.2306.12548

[5] Massimo Cencini, Fabio Ceccomi, Angelo Vulpiani, "Chaos", Series on Advances in Statistical Mechanics — Vol. 17, 2010

[6] Steven H. Strogatz, "NONLINEAR DYNAMICS AND CHAOS With Applications to Physics, Biology, Chemistry, and Engineering", 2015

[7] Henry D.I.Abarbanel, "Analys is of Observed Chaotic Data" , 1996

[8] Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning", 2016, http://www.deeplearningbook.org

[9] Henry D. I. Abarbanel, Reggie Brown, John J. Sidorowich, and Lev Sh. Tsimring, "The analysis of observed chaotic data in physical systems", Rev. Mod. Phys. , Vol. 65, No. 4, October 1993

[10] Juan Carlos Rocha, Reinette Biggs, Garry D. Peterson, "Regime Shifts: What are they and why do they matter?", 2014

[11] Miguel A. Munoz, "Colloquium: Criticality and dynamical scaling in living systems", May 2018

[12] Vittorio Loreto, Vito D. P. Servedio, Steven H. Strogatz and Francesca Tria, "Dynamics on expanding spaces: modeling the emergence of novelties", January 2017, https://arxiv.org/pdf/1701.00994

[13] Ling Feng, Lin Zhang, Choy Heng Lai, "Optimal Machine Intelligence at the Edge of Chaos", October 2020

[14] B.A. Huberman, T. Hogg, "Complexity and Adaptation", November 1986

[15] Stuart A. Kauffman, "Investigations", September 1996

[16] Robert M. French, "Catastrophic forgetting in connectionist networks", April 1999

[17] Nils Bertschinger, Thomas Natschlager, "Real Time Computation at the Edge of Chaos in Recurrent Neural Networks", January 2004

[18] Seolmin Yang, Hyejin Youn, "Geometrics of the Adjacent Possible: Harvesting Values at the Curvature", June 2024

[19] Razvan Pascanu, Tomas Mikolov, Yoshua Bengio , "Understanding the exploding gradient problem", November 2012

[20] Yoshua Bengio, Patrice Simard, Paolo Frasconi , "Learning Long-Term Dependencies with Gradient Descent is Difficult", March 1994

[21] Brian Chmiel, Liad Ben-Uri, Moran Shkolnik, Elad Hoffer, Ron Banner, Daniel Soudry, "Neural gradients are near-lognormal: improved quantized and sparse training", 2021

[22] German I. Parisi , Ronald Kemker, Jose L. Part, Christopher Kanan, Stefan Wermter , "Continual lifelong learning with neural networks: A review", July 2018

[23] James Kirkpatricka, Razvan Pascanua, Neil Rabinowitza, Joel Venessa, Guillaume Desjardinsa,Andrei A. Rusua, Kieran Milana, John Quana, Tiago Ramalhoa, Agnieszka Grabska-Barwinska a,Demis Hassabisa, Claudia Clopathb, Dharshan Kumarana, Raia Hadsella, "Overcoming catastrophic forgetting in neural networks", Jenuary 2017, https://arxiv.org/pdf/1612.00796

[24] Christof Teuscher, "Revisiting the edge of chaos: Again?", August 2022