Asymptotic Stability in Reservoir Computing

Jonathan Dong*, Erik Börve*, Mushegh Rafayelyan†, and Michael Unser*
*Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne, Lausanne 1015, Switzerland
†PhotonicsAI lab, Department of Physics, Yerevan State University, Yerevan, Armenia

2022 International Joint Conference on Neural Networks (IJCNN) | 978-1-7281-8671-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/IJCNN55064.2022.9892302

***Abstract*—Reservoir Computing is a class of Recurrent Neural Networks with internal weights fixed at random. Stability relates to the sensitivity of the network state to perturbations. It is an important property in Reservoir Computing as it directly impacts performance. In practice, it is desirable to stay in a stable regime, where the effect of perturbations does not explode exponentially, but also close to the chaotic frontier where reservoir dynamics are rich. Open questions remain today regarding input regularization and discontinuous activation functions. In this work, we use the recurrent kernel limit to draw new insights on stability in reservoir computing. This limit corresponds to large reservoir sizes, and it already becomes relevant for reservoirs with a few hundred neurons. We obtain a quantitative characterization of the frontier between stability and chaos, which can greatly benefit hyperparameter tuning. In a broader sense, our results contribute to understanding the complex dynamics of Recurrent Neural Networks.**

I. Introduction

Recurrent neural networks (RNN) represent a broad class of artificial neural networks. They present a temporal evolution with nonlinear internal dynamics and feedback loops, and are closer to biological neural networks compared to feedforward architectures. As such, their behavior is richer but harder to characterize. In particular, training RNNs remains a challenging problem and a topic of intense study [1]. For example, backpropagation typically exhibits exploding or vanishing gradients which limit our ability to train such networks.

To bypass this issue, reservoir computing (RC) fixes internal weights randomly and only adjusts the output weights of the network [2, 3]. This greatly facilitates the training process since one only has to learn a linear output model, with no problem of exploding or vanishing gradients. The philosophy behind RC is to use a large ensemble of randomly-connected neurons—the so-called *reservoir*—to embed time-dependent information in a way such that a linear mapping to the desired output is possible. RC has been applied to different areas such as speech recognition [4], chaos cryptography [5], and robot motor control [6]. It has been particularly promising for chaotic time series prediction [7].

The reservoir is a nonlinear dynamical system driven by an external input. Its properties need to be finely tuned to achieve optimal performance. On one hand, two distinct input patterns should lead to different reservoir states to distinguish them. On the other hand, oversensitivity can also be detrimental as the reservoir can fall in a chaotic dynamical regime, where perturbations explode exponentially with time. It is thus important to have a stable reservoir, in which information about the current reservoir state and input vanishes exponentially at a controllable rate. This necessary condition has been stated in the founding paper [2] as the *Echo-State Property*. The study of this transition between stable and chaotic regimes is enabled by the simplicity of RC, and we believe it may be relevant for other RNN architectures as well.

This stability property depends on hyperparameters of the reservoir—in particular, the standard deviation of the random and input weights. In practice, the best performance is typically obtained at the *edge of chaos*, a stable regime close to the chaotic frontier where dynamics are richer [8]. Basic observations have been proposed to help with hyperparameter tuning. For instance, if the activation function is 1-Lipschitz continuous, then stability for any input is achieved when the largest singular value of the internal weight matrix is smaller than one [9]. However, this bound is typically too conservative and a heuristic hyperparameter search is necessary for optimal performance.

This transition between stability and chaos raises several questions. Quantitative results are lacking for a broader class of activation functions like the rectified linear unit (ReLU, which is not differentiable at zero) or discontinuous activation functions. Moreover, it has been observed that the input regularizes the internal dynamics and allows for the use of spectral radii slightly larger than one [8]. A broader stability analysis would be beneficial since RC has been implemented in a large variety of settings [10, 11, 12, 13, 14]; an example being physical implementations with binary activation functions [15, 16].

Such questions are easier to tackle in the asymptotic limit, when the reservoir size is very large. In this large size limit, RC tends to a deterministic kernel that we iterate recurrently, called a recurrent kernel (RK) [17, 18]. The powerful interpretation of RK enables a mean-field study of stability. However, it has only been applied to reproduce known results with continuous activations and no input [17]. In another study, local Lyapunov exponents have been introduced to describe stability [9]. They provide a quantitative analysis of the stability of RC in the presence of an input. However, this metric needs to be computed as we iterate the reservoir, making it computationally-demanding for applications such as hyperparameter search. Moreover, it cannot handle non-differentiable functions.

In this paper, we show how this asymptotic limit enables us to quantitatively characterize stability in the presence of an input and with discontinuous activation functions. In particular, we exhibit two important properties of the activation function impacting stability—continuity and boundedness. This kernel limit greatly facilitates quantitative studies as stability boils

978-1-7281-8671-9/22/$31.00 ©2022 European Union

down to analyzing fixed points iterations of deterministic functions.

In Section 2, we give a basic description of RC and RK, along with a proper definition of stability in both cases. In Section 3, we then show how to apply this stability study to three representative examples: the error function activation (bounded and continuous), the sign function (bounded and discontinuous), and the Rectified Linear Unit or ReLU (unbounded and continous).

II. Theoretical Background

A. Reservoir Computing

1) Definition: In the class of RC, we focus on the Echo-State Networks. These generic randomly-connected RNNs were first introduced by Jaeger [2] and are the most commonly-used ones in the field. A time-dependent input $\mathbf{i}^{(t)} \in \mathbb{R}^d$ is fed into a reservoir of size N, yielding the following update equation for the reservoir state $\mathbf{x}^{(t)} \in \mathbb{R}^N$:

$$\mathbf{x}^{(t+1)} = \frac{1}{\sqrt{N}} f(\mathbf{W}_r \mathbf{x}^{(t)} + \mathbf{W}_i \mathbf{i}^{(t)}). \tag{1}$$

Here, f is an element-wise activation function, $\mathbf{W}_r$ the internal reservoir weights, and $\mathbf{W}_i$ the input weights. These weights are drawn from i.i.d. distributions $\mathcal{N}(0, \sigma_r^2)$ and $\mathcal{N}(0, \sigma_i^2)$, respectively. To emphasize the particularity of RC, these weights are fixed randomly and are not trained. Thus, iterating this update equation only depends on the input data.

The algorithm generates an output $\mathbf{o}^{(t)}$ using a linear model

$$\mathbf{o}^{(t)} = \mathbf{W}_o \mathbf{x}^{(t)}. \tag{2}$$

This training step consists of a simple linear regression. In RC, the complexity of the computation is performed during the nonlinear update described in Eq. (1).

Current research directions include the physical implementation of RC on dedicated hardware. Thanks to the flexibility of RC, optical devices, dedicated electronics, and more exotic architectures have been proposed for small-footprint and fast computation. On top of that, recent software developments include Deep Reservoir Computing [19, 20], in which a hierarchical architecture has been proposed to improve performance.

2) Stability: As discussed previously, stability is a fundamental property to study in RC. This stability is typically characterized by the experiment depicted in Fig. 1a. We initialize two different reservoirs $\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}$ independently from i.i.d. Gaussian distributions. They share the same weights $\mathbf{W}_r$ and $\mathbf{W}_i$ and receive the same input $\mathbf{i}^{(t)}$. At each time step, the input $\mathbf{i}^{(t)} \in \mathbb{R}^d$ is randomly drawn from the unit sphere. We investigate whether these reservoirs converge towards a common trajectory.

The stability metric we choose quantifies the distance between the two reservoir states as it evolves with time. It is given by

$$L^{(t)} = \left\| \mathbf{x}_1^{(t)} - \mathbf{x}_2^{(t)} \right\|^2. \tag{3}$$

A reservoir configuration is called *stable* for input $\mathbf{i}^{(t)}$ if $\lim_{t\to\infty} L^{(t)} = 0$. Conversely, the Echo-State Property does not hold when $\lim_{t\to\infty} L^{(t)} > 0$. This is an indirect characterization of chaos; other behaviors such as several distinct fixed points would also be possible but both settings would not be suitable for a Reservoir Computing algorithm.

The stability property depends on how the weights are set. The two parameters σ_r^2 and σ_i^2 will tune the transition between stability and chaos, by changing the variance of the random weights. Small internal weights exponentially damp the importance of previous reservoir states, while large internal weights tend to increase initial perturbations leading to a chaotic behavior.

Stability has been characterized for Lipschitz-continuous functions for a random connectivity matrix [9]. For example, since erf is $2/\sqrt{\pi}$-Lipschitz, stability is ensured when $\sigma_r < \sqrt{\pi}/2$. This result stands for any input and is optimal when there is zero input. Nevertheless, it is too conservative when an input is present.

B. Recurrent Kernels

1) Definition: As a linear model after a non-linear embedding, RC has tight links with kernel methods. This class of algorithms implicitly performs a linear regression in the embedding space based on scalar products between pairs of points. Kernels have been studied extensively and, in particular, Random Features have been proposed as finite-dimensional approximations of kernels.

Similarly, the limit of RC when N goes to infinity is defined as an RK. RC can be interpreted as the temporal equivalent of Random Features: a finite-dimensional approximation of a deterministic RK.

More precisely, this RK operates on time-dependent scalar products between two reservoir states $\mathbf{x}^{(t)}$ and $\mathbf{y}^{(t)}$ driven by inputs $\mathbf{i}^{(t)}$ and $\mathbf{j}^{(t)}$, respectively. Let $\mathbf{w}_{r,j}$ and $\mathbf{w}_{i,j}$ be the j-th rows of $\mathbf{W}_r$ and $\mathbf{W}_i$ respectively. Eq. (1) then yields

$$\langle \mathbf{x}^{(t+1)}, \mathbf{y}^{(t+1)} \rangle = \frac{1}{N} \sum_j f(\mathbf{w}_{r,j}^\top \mathbf{x}^{(t)} + \mathbf{w}_{i,j}^\top \mathbf{i}^{(t)}) \times f(\mathbf{w}_{r,j}^\top \mathbf{y}^{(t)} + \mathbf{w}_{i,j}^\top \mathbf{j}^{(t)}). \tag{4}$$

This sum of independent random terms concentrates like Random Features of a certain kernel k [21]:

$$\langle \mathbf{x}^{(t+1)}, \mathbf{y}^{(t+1)} \rangle \to k\left(\begin{bmatrix} \mathbf{x}^{(t)} \\ \mathbf{i}^{(t)} \end{bmatrix}, \begin{bmatrix} \mathbf{y}^{(t)} \\ \mathbf{j}^{(t)} \end{bmatrix} \right), \tag{5}$$

i.e. a kernel on the concatenation of reservoirs and inputs. This kernel function k is defined by the activation f and the distribution of the random vectors $p(\mathbf{w})$. To define deterministic RKs which are not linked to a particular RC algorithm, one needs to remove the dependence in previous embeddings $\mathbf{x}^{(t)}$ and $\mathbf{y}^{(t)}$. This is possible for any rotationally-invariant distribution $p(\mathbf{w})$, a more general setting than [18].

To summarize, RKs are deterministic algorithms where the input data $\mathbf{i}^{(t)}$ is used to recurrently update a Gram matrix $\mathbf{G}^{(t)}$, the matrix containing scalar products between all pairs of points. This Gram matrix is initialized arbitrarily $\mathbf{G}^{(0)}$ and updated as

$$\mathbf{G}^{(t+1)} = k(\mathbf{G}^{(t)}, \{\langle \mathbf{i}^{(t)}, \mathbf{j}^{(t)} \rangle\}_{\mathbf{i},\mathbf{j}}), \tag{6}$$

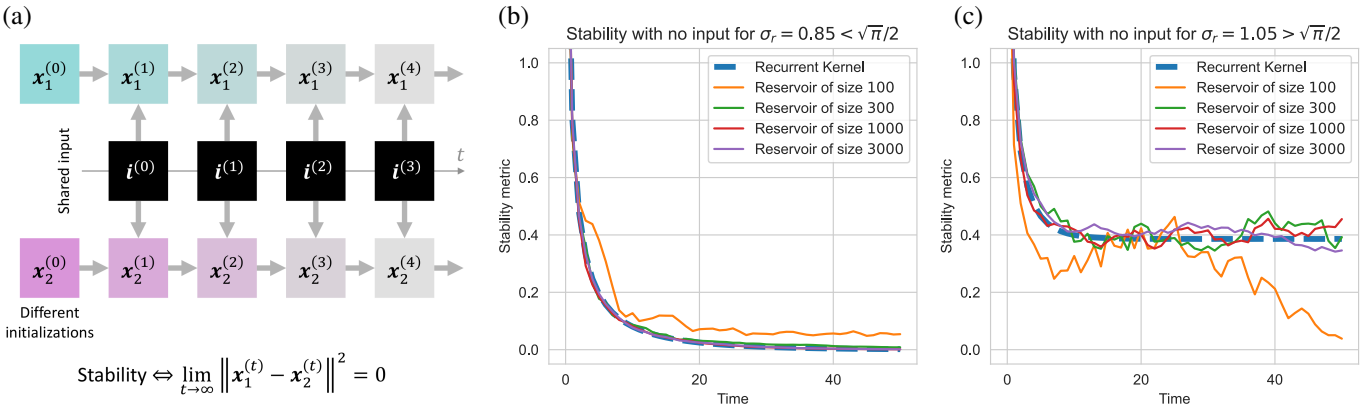

Fig. 1. Principle and motivation for an asymptotic stability study. **(a) Scheme of a stability test.** To study stability in Reservoir Computing, two identical reservoirs are initialized differently and driven by the same input. The reservoir is stable if after a transient time, the reservoir state does not depend on this arbitrary initialization. We thus monitor the squared distance between the reservoir states through time. **(b-c) Asymptotic stability study in stable and chaotic cases.** Stability metric $L^{(t)}$ as a function of time t for various reservoir sizes and for the corresponding Recurrent Kernel limit. The stable case corresponds to an erf activation function with $\sigma_r = 0.85$ and $\sigma_i = 0$. The chaotic case corresponds to the same previous parameters with the exception of $\sigma_r = 1.05$.

where $\{\langle \mathbf{i}^{(t)}, \mathbf{j}^{(t)} \rangle\}_{\mathbf{i},\mathbf{j}}$ denotes the scalar products between all pairs of inputs at time t. For conciseness, we use the same letter k for the kernel, a more detailed derivation for common kernels can be found in [18].

Convergence of RC towards its RK limit has been observed in practice in a large range of settings, when the activation function is bounded. Formally, convergence has only been proven in restrictive settings [18], which is why convergence will be assessed on a case-by-case basis.

In practice, computing directly with the RK limit is beneficial for medium-sized tasks with a few thousand examples. Iterating them is efficient since it mostly consists of element-wise operations. Kernel methods typically struggle when the number of training points becomes very large as they need to compute scalar products between all pairs of points.

2) Stability: It is then natural to describe with RKs the limit of this stability metric as $N \to \infty$. The two reservoirs evolving in parallel define a 2×2 Gram matrix

$$\mathbf{G}_N^{(t)} = \begin{pmatrix} \left\| \mathbf{x}_1^{(t)} \right\|^2 & \left\langle \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \right\rangle \\ \left\langle \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \right\rangle & \left\| \mathbf{x}_2^{(t)} \right\|^2 \end{pmatrix}. \tag{7}$$

This matrix is symmetric and invariant by permutation of $\mathbf{x}_1$ and $\mathbf{x}_2$. For this reason, we introduce $G_{\text{eq}}^{(t)} = \|\mathbf{x}_1^{(t)}\|^2 = \|\mathbf{x}_2^{(t)}\|^2$ and $G_{\text{neq}}^{(t)} = \langle \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \rangle$.

Since the reservoirs are initialized with independent random draws, the limit at $t = 0$ when $N \to \infty$ is the identity matrix:

$$\mathbf{G}^{(0)} = \lim_{N\to\infty} \mathbf{G}_N^{(0)} = \mathbf{I}_2. \tag{8}$$

This defines the initial state of our recurrent kernel. Equivalently, $G_{\text{eq}}^{(0)} = 1$ and $G_{\text{neq}}^{(0)} = 0$.

We then iterate this recurrent kernel with the input $\mathbf{i}^{(t)}$. The stability metric defined for RC in Eq. (3) has a RK equivalent, simply obtained by developing the squared norm,

$$\mathcal{L}^{(t)} = 2(G_{\text{eq}}^{(t)} - G_{\text{neq}}^{(t)}). \tag{9}$$

This simple expression comes from the deterministic nature of RKs. We only have to compute how the two scalar quantities evolve with time instead of distances between high-dimensional vectors. This will enable us to perform analytic studies to quantify the input regularization for example or to tackle the case of discontinuous activation functions.

3) Examples: In this work, we will use three different activation functions, all with Gaussian random weights. The error function $f_1 = \text{erf}$ corresponds to an arcsine kernel in Eq. (5)

$$k_1(\mathbf{u}, \mathbf{v}) = \frac{2}{\pi} \arcsin\left(\frac{2\langle \mathbf{u}, \mathbf{v} \rangle}{\sqrt{(1 + 2\|\mathbf{u}\|^2)(1 + 2\|\mathbf{v}\|^2)}} \right), \tag{10}$$

the sign function $f_2 = \text{sign}$ to

$$k_2(\mathbf{u}, \mathbf{v}) = \frac{2}{\pi} \arcsin\left(\frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\|\mathbf{u}\|\|\mathbf{v}\|} \right), \tag{11}$$

and the Rectified Linear Unit $f_3 = \text{ReLU}$ to

$$k_3(\mathbf{u}, \mathbf{v}) = \frac{1}{2\pi} \left(\langle \mathbf{u}, \mathbf{v} \rangle \arccos\left(-\frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\|\mathbf{u}\|\|\mathbf{v}\|} \right) + \sqrt{\|\mathbf{u}\|^2 \|\mathbf{v}\|^2 - \langle \mathbf{u}, \mathbf{v} \rangle^2} \right). \tag{12}$$

To link with our case of RKs, $\mathbf{u}$ and $\mathbf{v}$ correspond respectively to $\begin{bmatrix} \sigma_r \mathbf{x}^{(t)} \\ \sigma_i \mathbf{i}^{(t)} \end{bmatrix}$ and $\begin{bmatrix} \sigma_r \mathbf{y}^{(t)} \\ \sigma_i \mathbf{j}^{(t)} \end{bmatrix}$ in Eq. (5). These three activation functions have been chosen as they are representative of the diversity between bounded and unbounded functions, as well as Lipschitz-continuous and discontinuous functions.

III. Results

A. Convergence of RC towards RK

For the classical case of an erf activation function with no input, we know that stability occurs when $\sigma_r < \sqrt{\pi}/2$ [9]. In Fig 1b and 1c, we display the time evolution of the stability metrics $L^{(t)}$ and $\mathcal{L}^{(t)}$ for $\sigma_r = 0.85 < \sqrt{\pi}/2$ and $\sigma_r = 1.05 >$

$\sqrt{\pi}/2$ respectively. Indeed, we observe that in the first case, the stability metric converges to 0 as t increases, whereas it does not converge to zero in the chaotic regime.

We see in these results the convergence of RC to the RK limit. As the reservoir size increases, its stability metric $L^{(t)}$ tends to the well-defined kernel limit $\mathcal{L}^{(t)}$. This motivates our asymptotic stability analysis. In practice, our RK limit accurately describes stability of RC even for reservoirs of moderate sizes around a few hundreds. Finite size effects may appear for smaller sizes and mostly in the chaotic regime. We can thus leverage this deterministic update equation to study stability for various activation functions in the presence of an input.

In the following, we show exact results for this RK limit. It will describe typical behaviors of RC as well, neglecting finite-size effects.

B. Erf activation function

Eq. (9) defines two sequences $G_{\text{eq}}^{(t)}$ and $G_{\text{neq}}^{(t)}$, initialized with $G_{\text{eq}}^{(0)} = 1$ and $G_{\text{neq}}^{(0)} = 0$, with update equations deduced from Eq. (10) for the erf activation function:

$$\begin{cases} G_{\text{eq}}^{(t+1)} = \dfrac{2}{\pi} \arcsin\left(\dfrac{2\sigma_r^2 G_{\text{eq}}^{(t)} + 2\sigma_i^2}{1 + 2\sigma_r^2 G_{\text{eq}}^{(t)} + 2\sigma_i^2} \right) \\ G_{\text{neq}}^{(t+1)} = \dfrac{2}{\pi} \arcsin\left(\dfrac{2\sigma_r^2 G_{\text{neq}}^{(t)} + 2\sigma_i^2}{1 + 2\sigma_r^2 G_{\text{eq}}^{(t)} + 2\sigma_i^2} \right) \end{cases} \tag{13}$$

Thanks to our asymptotic model, we can precisely characterize the transition between stability and chaos. The full derivation is detailed in Appendix B.

Proposition 1. *The two sequences G_{eq} and G_{neq} are convergent. For any $\sigma_r \geq \sqrt{\pi}/2$, the frontier between stability and chaos is given by:*

$$\psi(\sigma_r) = \frac{4\sigma_r^2}{\pi} - \frac{1}{4} - \frac{2\sigma_r^2}{\pi} \arcsin\left(\frac{16\sigma_r^4 - \pi^2}{16\sigma_r^4 + \pi^2} \right). \tag{14}$$

- *If $\sigma_i \geq \psi(\sigma_r)$, the dynamics are stable.*
- *If $\sigma_i < \psi(\sigma_r)$, the dynamics are chaotic.*

Reciprocally, since ψ is a bijection from $[\sqrt{\pi}/2, +\infty)$ to $[0, +\infty)$, for any $\sigma_i \geq 0$:

- *If $\sigma_r \leq \psi^{-1}(\sigma_i)$, the dynamics are stable.*
- *If $\sigma_r > \psi^{-1}(\sigma_i)$, the dynamics are chaotic.*

We display in Fig. 2a, the limit of $\mathcal{L}^{(t)}$ as a function of the hyperparameters σ_r and σ_i. This limit is equal to 0 for small values of σ_r. This corresponds to the region in which the reservoir is in a stable regime. The limit of $\mathcal{L}^{(t)}$ becomes non-zero for large values of σ_r, which indicates chaotic reservoir dynamics. The frontier between the stable and chaotic regions depends on σ_i, the standard deviation of the input weights. Having a large input pushes the transition between stability and chaos to larger values of σ_r.

Without an input, i.e. for $\sigma_i = 0$, we obtain $\psi^{-1}(0) = \sqrt{\pi}/2$. We thus recover the previous result which is optimal with no input [17]. Additionally, we quantify how much the input regularizes the dynamics. This comes from the saturation of the activation function. With a large input, the arguments u of the activations $\text{erf}(u)$ are typically larger and we are in the flatter regions of the activation function. The network is therefore less sensitive to changes.

The quantitative characterization of the frontier may provide a useful tool to restrict the hyperparameter search space. As we want to stay close to this frontier between stability and chaos for optimal performance, it transforms a two-dimensional hyperparameter search on both (σ_i, σ_r) to a unidimensional search along the frontier.

C. Sign activation function

The equations to update the two quantities in Eq. (9) is deduced from Eq. (11) for the sign activation function:

$$\begin{cases} G_{\text{eq}}^{(t+1)} &= 1 \\ G_{\text{neq}}^{(t+1)} &= \dfrac{2}{\pi} \arcsin\left(\dfrac{\sigma_r^2 G_{\text{neq}}^{(t)} + \sigma_i^2}{\sigma_r^2 + \sigma_i^2} \right) \end{cases} \tag{15}$$

Proposition 2. *As soon as $\sigma_r > 0$ and for any value σ_i, the stability metric is converging to a non-zero value, i.e.*

$$\lim_{t \to \infty} \mathcal{L}^{(t)} = l > 0. \tag{16}$$

This implies that any reservoir with sign activations is chaotic.

The case of discontinuous activation functions has not received a lot of attention before. In the asymptotic limit, there is an averaging effect that makes the kernel limit continuous. Despite this well-defined limit, there is no rigorous stability in the sense that $\mathcal{L}^{(t)}$ converges to 0 exactly.

Fig. 2b shows this limit l as a function of (σ_r, σ_i). We see that there is no stable region, apart from the trivial case $\sigma_r = 0$. However, similar to the saturation effect with erf, the addition of an input seems to regularize the dynamics. For large values of σ_i, it is possible to still control the stability of the system. More precisely, when $\sigma_i \gg \sigma_r$, we have:

$$l \approx \frac{16\sigma_r^2}{\pi^2 \sigma_i^2}. \tag{17}$$

In this case, the input regularizes the dynamics and $\lim_{\sigma_i \to \infty} l = 0$.

Indeed, training with step activation functions has been performed successfully in practice [15, 16]. This observation may be important for physical implementations of RC or low-power RC with quantized activation functions.

D. ReLU activation function

The equations to update the two quantities in Eq. (9) are deduced from Eq. (12) for the ReLU activation function:

$$\begin{cases} G_{\text{eq}}^{(t+1)} = & \dfrac{1}{2}\left(\sigma_r^2 G_{\text{eq}}^{(t)} + \sigma_i^2 \right) \\ G_{\text{neq}}^{(t+1)} = & \dfrac{1}{2\pi}\left(\sigma_r^2 G_{\text{neq}}^{(t)} + \sigma_i^2 \right) \arccos\left(-\dfrac{\sigma_r^2 G_{\text{neq}}^{(t)} + \sigma_i^2}{\sigma_r^2 G_{\text{eq}}^{(t)} + \sigma_i^2} \right) \\ & + \dfrac{1}{2\pi}\sqrt{\left(\sigma_r^2 G_{\text{eq}}^{(t)} + \sigma_i^2 \right)^2 - \left(\sigma_r^2 G_{\text{neq}}^{(t)} + \sigma_i^2 \right)^2} \end{cases} \tag{18}$$

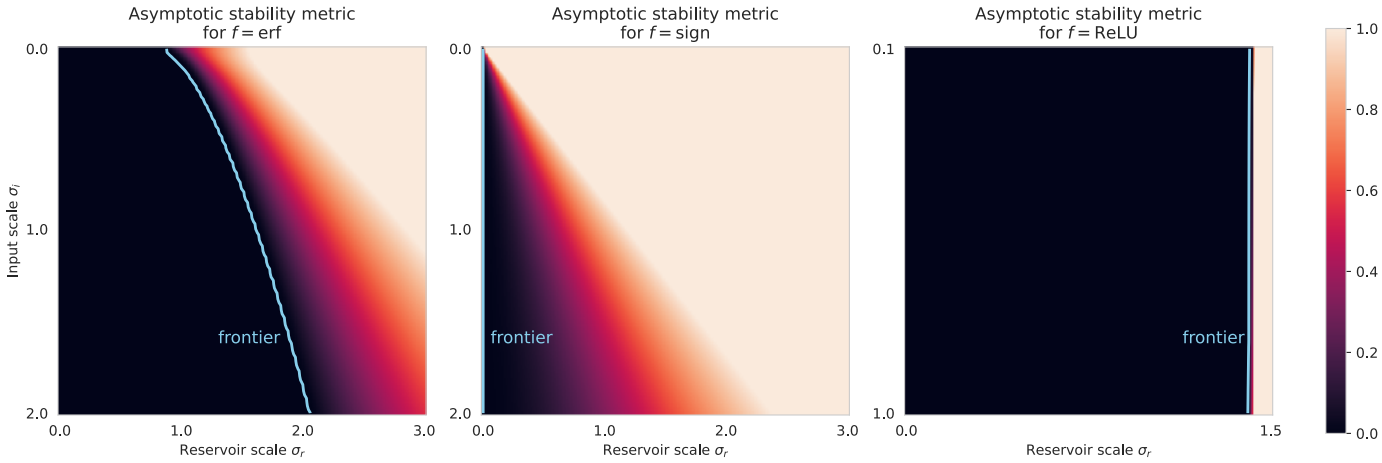

Fig. 2. **(a-c) Asymptotic stability metric for $f = \mathrm{erf}$, $f = \mathrm{sign}$, and $f = \mathrm{ReLU}$ as a function of σ_i and σ_r.** Asymptotic values are computed from the update equations of the recurrent kernel limit, for $t = 200$ large enough. When this stability metric converges to 0, the recurrent kernel dynamics are stable. Whenever it converges to a non-zero value, the recurrent kernel dynamics are unstable or chaotic. In blue is drawn the frontier between the stable and chaotic regions.

Proposition 3. *If $\sigma_r < \sqrt{2}$ and for any value σ_i, the RK is stable, i.e. we have*

$$\lim_{t\to\infty} \mathcal{L}^{(t)} = 0. \tag{19}$$

Fig. 2c shows this limit as a function of (σ_r, σ_i). We observe that there is no input regularization. In contrast with the erf case, the frontier shows no dependence on σ_i. This is linked with the absence of saturation in the ReLU activation function.

For $\sigma_r > \sqrt{2}$, the stability metric diverges and the RK is unstable. An interesting point to notice is that despite ReLU being 1-Lipschitz, the frontier is not for $\sigma_r = 1$ as it could have been predicted from classical analysis of the Lipschitz constant. Instead, slightly larger reservoir weights are possible, thanks to the subdifferentiability of ReLU at zero.

To apply these results to RC, particular care needs to be taken here regarding the convergence of RC towards its RK limit. This convergence is quite robust in practice for bounded activation functions but it is not always the case with ReLU activations. Indeed, we show in Appendix A that convergence is obtained in a large part of the stable region but not in the unstable region.

IV. Discussion

In this work, we have presented a framework to study the asymptotic stability of RC. We relied on the recurrent kernel limit to quantitatively characterize trajectories when the reservoir size is large. We then applied our framework to three different activation functions. We showed the importance of having a continuous activation function and made the link between input regularization and saturation of the activation function.

These results can be important in practice for hyperparameter tuning. They also help to develop a better understanding of stability in non-classical cases. We believe this framework is powerful enough to be applied to a wide range of applications.

In the future, more general results with strong convergence proofs of RC towards RKs may be derived, supporting the observations presented here. This study may be generalized to a larger class of functions, like the hyperbolic tangent which is commonly used in RC. The observed behaviors could be extended to any differentiable and saturating activation function. The corresponding kernel and related quantities may not have an analytic expression, but they can still be computed with integrals.

One may also extend this approach to other RC architectures such as Deep Reservoir Computing [19, 20]. As a more general comment, this kernel approach may be relevant for non-recurrent architectures as well, to understand better the propagation of perturbations in neural networks.

The associated code is available at [22].

Acknowledgements

We would like to thank Pakshal Bohra and Tony Wu for their insightful comments to revise this paper.

Funding

J.D. and M.U. acknowledge funding from European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 101020573 FunLearn).

References

[1] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. "On the difficulty of training recurrent neural networks". In: *International conference on machine learning*. 2013, pp. 1310–1318.

[2] Herbert Jaeger. "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note". In: *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report* 148.34 (2001), p. 13.

[3] David Verstraeten et al. "An experimental unification of reservoir computing methods". In: *Neural networks* 20.3 (2007), pp. 391–403.
[4] Laurent Larger et al. "High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification". In: *Physical Review X* 7.1 (2017), p. 011015.
[5] Piotr Antonik et al. "Using a reservoir computer to learn chaotic attractors, with applications to chaos synchronization and cryptography". In: *Physical Review E* 98.1 (2018), p. 012215.
[6] Mantas Lukoševičius, Herbert Jaeger, and Benjamin Schrauwen. "Reservoir computing trends". In: *KI-Künstliche Intelligenz* 26.4 (2012), pp. 365–371.
[7] Jaideep Pathak et al. "Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach". In: *Physical review letters* 120.2 (2018), p. 024102.
[8] Mantas Lukoševičius and Herbert Jaeger. "Reservoir computing approaches to recurrent neural network training". In: *Computer Science Review* 3.3 (2009), pp. 127–149.
[9] Gilles Wainrib and Mathieu N Galtier. "A local echo state property through the largest Lyapunov exponent". In: *Neural Networks* 76 (2016), pp. 39–45.
[10] Lennert Appeltant et al. "Information processing using a single dynamical node as complex system". In: *Nature communications* 2 (2011), p. 468.
[11] Laurent Larger et al. "Photonic information processing beyond Turing: an optoelectronic implementation of reservoir computing". In: *Optics express* 20.3 (2012), pp. 3241–3249.
[12] Michiel Hermans et al. "Photonic delay systems as machine learning implementations". In: (2015).
[13] Gouhei Tanaka et al. "Recent advances in physical reservoir computing: A review". In: *Neural Networks* (2019).
[14] Mushegh Rafayelyan et al. "Large-scale optical reservoir computing for spatiotemporal chaotic systems prediction". In: *Physical Review X* 10.4 (2020), p. 041037.
[15] Jonathan Dong et al. "Scaling up Echo-State Networks with multiple light scattering". In: *2018 IEEE Statistical Signal Processing Workshop (SSP)*. IEEE. 2018, pp. 448–452.
[16] Jonathan Dong et al. "Optical Reservoir Computing using multiple light scattering for chaotic systems prediction". In: *IEEE Journal of Selected Topics in Quantum Electronics* 26.1 (2019), pp. 1–12.
[17] Michiel Hermans and Benjamin Schrauwen. "Recurrent kernel machines: Computing with infinite echo state networks". In: *Neural Computation* 24.1 (2012), pp. 104–133.
[18] Jonathan Dong et al. "Reservoir Computing meets Recurrent Kernels and Structured Transforms". In: *Advances in Neural Information Processing Systems* 33 (2020).
[19] Claudio Gallicchio, Alessio Micheli, and Luca Pedrelli. "Deep reservoir computing: A critical experimental analysis". In: *Neurocomputing* 268 (2017), pp. 87–99.
[20] Claudio Gallicchio and Simone Scardapane. "Deep Randomized Neural Networks". In: *Recent Trends in Learning From Data*. Springer, 2020, pp. 43–68.
[21] Ali Rahimi and Benjamin Recht. "Random features for large-scale kernel machines". In: *Advances in neural information processing systems*. 2008, pp. 1177–1184.
[22] Jonathan Dong and Erik Börve. https://github.com/jon-dong/recurrent-kernel-stability. 2022.

APPENDIX A
CONVERGENCE OF RC TOWARDS RK

The results shown in Fig. 2 are exact for RKs, but convergence of RC towards the RK limit is required to translate them in RC. This convergence needs to be assessed for each combination of hyperparameters (σ_i, σ_r).

We evaluate this convergence by iterating both a reservoir of size $N = 2000$ and the associated RK for each activation function presented previously. They are fed two random input time series $\mathbf{i}^{(t)}$ and $\mathbf{j}^{(t)}$ of length 50 and the final 2×2 Gram matrices are computed, denoted G for RC and $\mathcal{G}$ for the associated RK. Our convergence metric is defined as

$$E = \|G - \mathcal{G}\|_F^2 \tag{20}$$

with $\|\cdot\|_F$ the Frobenius norm.

We see in Fig. 3 that convergence is reliably obtained with erf and sign activation functions. On the other hand, with an unbounded ReLU activation function, convergence does not happen for large values of σ_r. This implies that the previous theorems also hold for RC, apart from the ReLU case for large σ_r. This apparent link between stability and convergence of RC towards a kernel limit calls for more investigation.

APPENDIX B
TECHNICAL RESULTS FOR $f = \mathrm{erf}$

A. Study of G_{eq}

We define the function h_1 on $[0, 1]$ by

$$h_1 : x \mapsto \frac{2}{\pi} \arcsin\left(\frac{2\sigma_r^2 x + 2\sigma_i^2}{1 + 2\sigma_r^2 x + 2\sigma_i^2}\right) \tag{21}$$

such that $G_{\mathrm{eq}}^{(t+1)} = h_1\left(G_{\mathrm{eq}}^{(t)}\right)$.

Since $h_1(0) = \frac{2}{\pi}\arcsin\left(\frac{2\sigma_i^2}{1+2\sigma_i^2}\right) > 0$, $h_1(1) = \frac{2}{\pi}\arcsin\left(1 - \frac{1}{1+2\sigma_r^2+2\sigma_i^2}\right) < 1$, and the continuity of h_1, the intermediate value theorem ensures the existence of at least one fixed point for h_1.

Moreover, h_1 is strongly concave, as it is twice differentiable with

$$h_1''(x) = -\frac{16\sigma_r^4(1 + 3\sigma_r^2 x + 3\sigma_i^2)}{\pi(1 + 2\sigma_r^2 x + 2\sigma_i^2)^2(1 + 4\sigma_r^2 x + 4\sigma_i^2)^{3/2}} < 0 \tag{22}$$

for $x \in [0, 1]$. h_1 thus has at most two fixed points.

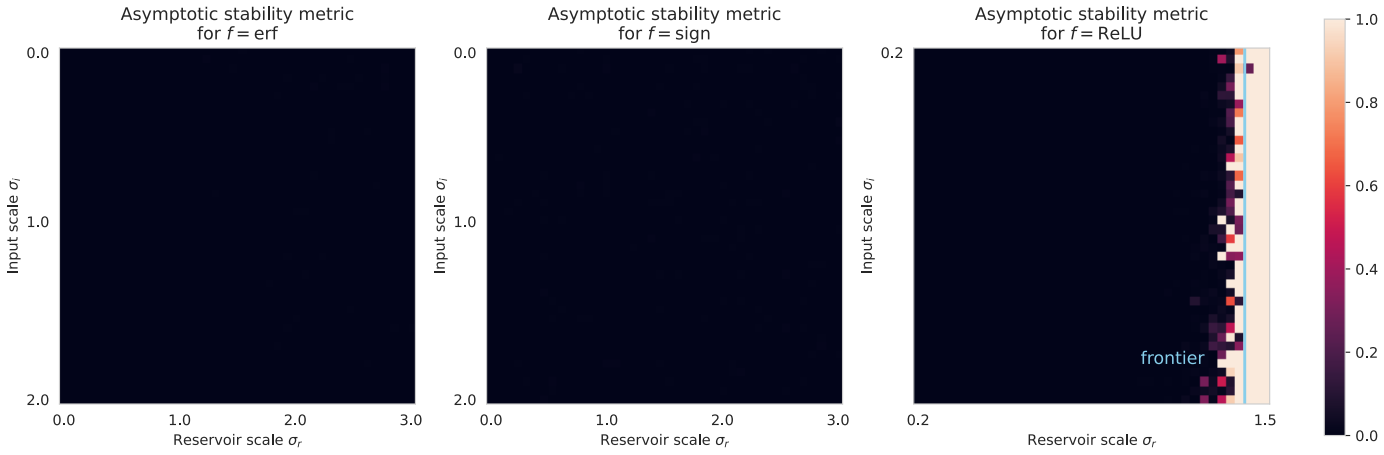

Fig. 3. **(a-c) Convergence study of RC towards RK for $f = \mathrm{erf}$, $f = \mathrm{sign}$, and $f = \mathrm{ReLU}$ as a function of σ_i and σ_r.** Frobenius norm E between the final Gram matrices obtained from a reservoir of size $N = 2000$ and the Recurrent Kernel equivalent, iterated with two different random inputs $\mathbf{i}^{(t)}$ and $\mathbf{j}^{(t)}$ of length 50. We observe robust convergence when the activation function is bounded or for small values of σ_r.

If h_1 had two fixed points $a_1 < a_2$, then the strong concavity of h_1 would imply $h_1(0) < a + \frac{b-a}{b-a}(0-a) = 0$, which contradicts the first observation of this proof. Thus, h_1 has a unique fixed point that we denote by a.

Since $h_1(1) < 1$, $h_1(x) \neq x$ for $x \in (a, 1]$, and because h_1 is continuous, we necessarily have $h_1(x) < x$ for all $x \in (a, 1]$. The sequence G_{eq} is thus decreasing. As it is non-negative, thus bounded below, it converges to a, the fixed point of h_1.

B. Study of G_{neq}

We define the function h_2^g, defined on $[0, a]$ and parametrized by $g \in [a, 1]$, by

$$h_2^g : x \mapsto \frac{2}{\pi} \arcsin\left(\frac{2\sigma_r^2 x + 2\sigma_i^2}{1 + 2\sigma_r^2 g + 2\sigma_i^2}\right) \tag{23}$$

such that $G_{\text{neq}}^{(t+1)} = h_2^{G_{\text{eq}}^{(t)}}\left(G_{\text{neq}}^{(t)}\right)$. As G_{eq} converges to a, we will study the sequence g_{neq} defined by $g_{\text{neq}}^{(0)} = 0$ and $g_{\text{neq}}^{(t+1)} = h_2^a\left(g_{\text{neq}}^{(t)}\right)$.

Since h_2^a is strongly convex as it is of the form $h_2^a(x) = A \arcsin(Bx + C)$ with $A, B, C > 0$, it has at most two fixed points. One of them is a, since

$$h_2^a(a) = \frac{2}{\pi} \arcsin\left(\frac{2\sigma_r^2 a + 2\sigma_i^2}{1 + 2\sigma_r^2 a + 2\sigma_i^2}\right) = h_1(a) = a. \tag{24}$$

Let b be the smallest fixed point of h_2^a. Because h_2^a is an increasing non-negative function, $0 \leq h_2^a(x) \leq b$ and g_{neq} stays in $[0, b]$.

$h_2^a(x) \neq x$ for $x < b$ by definition of b. As $h_2^a(0) > 0$ and using the continuity of h_2^a, $h_2^a(x) > x$ for $x \in [0, b)$. g_{neq} is hence an increasing sequence. Because it is bounded above, it converges to b, the unique fixed point of h_2^a restricted to $[0, a]$.

C. Equation of the frontier

Using Eq. (9), the limit of $\mathcal{L}^{(t)}$ is

$$\lim_{t \to \infty} \mathcal{L}^{(t)} = 2(a - b). \tag{25}$$

We want to study when a is the smallest fixed point of h_2^a.

This property is linked with the derivative $(h_2^a)'(a)$. Since h_2^a is strongly convex with $h_2^a(0) > 0$, if $(h_2^a)'(a) > 1$, there exists another fixed point in $(0, a)$, while when $(h_2^a)'(a) < 1$, a is the smallest fixed point of h_2^a. The derivative is given by

$$(h_2^a)'(g) = \frac{4\sigma_r^2}{\pi\sqrt{1 + 4\sigma_r^2 g + 4\sigma_i^2}}. \tag{26}$$

The frontier corresponds to the equation $(h_2^a)'(a) = 1$. This equation can be rewritten as:

$$a = \frac{4\sigma_r^2}{\pi^2} - \frac{1}{4\sigma_r^2} - \frac{\sigma_i^2}{\sigma_r^2}. \tag{27}$$

Injecting this in the equation defining a as a fixed point of h_1, i.e.

$$a = \frac{2}{\pi} \arcsin\left(\frac{2\sigma_r^2 a + 2\sigma_i^2}{1 + 2\sigma_r^2 a + 2\sigma_i^2}\right), \tag{28}$$

yields the desired equation for the frontier.

Appendix C
Technical results for $f = \mathrm{sign}$

We define the function h_2 on $(0, 1)$ by

$$h_2 : x \mapsto \frac{2}{\pi} \arcsin\left(1 - \frac{\sigma_r^2(1 - x)}{\sigma_r^2 + \sigma_i^2}\right), \tag{29}$$

such that $G_{\text{neq}}^{(t+1)} = h_2\left(G_{\text{neq}}^{(t)}\right)$. G_{neq} corresponds to fixed point iteration of h_2, starting at $G_{\text{neq}}^{(0)} = 0$. It therefore converges to the smallest fixed point of h_2.

h_2 is strictly convex, and thus has at most two fixed points. $x = 1$ corresponds to one such fixed point with a vertical tangent. Since $h_2(0) > 0$, there is another fixed point in $(0, 1)$, that we denote by b. In the end, $\lim_{t \mapsto \infty} \mathcal{L}^{(t)} = 2(1 - b) > 0$.

For $\sigma_i \gg \sigma_r$, the asymptotic approximation of $\arcsin$ gives

$$1-b = 1-h_2(b) \tag{30}$$

$$= 1-\frac{2}{\pi}\arcsin\left(1-\frac{\sigma_r^2(1-b)}{\sigma_i^2}+O\left(\frac{\sigma_r^4}{\sigma_i^4}\right)\right) \tag{31}$$

$$= 1-\frac{2}{\pi}\left(\frac{\pi}{2}-\sqrt{2}\sqrt{\frac{\sigma_r^2(1-b)}{\sigma_i^2}}\right)+O\left(\frac{\sigma_r^3}{\sigma_i^3}\right) \tag{32}$$

$$= \frac{2\sqrt{2}\sigma_r\sqrt{1-b}}{\pi\sigma_i}+O\left(\frac{\sigma_r^3}{\sigma_i^3}\right). \tag{33}$$

Taking the square, we finally obtain

$$1-b = \frac{8\sigma_r^2}{\pi^2\sigma_i^2}+O\left(\frac{\sigma_r^3}{\sigma_i^3}\right). \tag{34}$$

APPENDIX D
TECHNICAL RESULTS FOR $f = \mathrm{ReLU}$

We define the function h_1 on $\mathbb{R}_+$ by

$$h_1 : x \mapsto \frac{1}{2}\left(\sigma_r^2 x+\sigma_i^2\right). \tag{35}$$

We have $G_{\text{eq}}^{(t+1)} = h_1\left(G_{\text{eq}}^{(t)}\right)$. Since h_1 is an affine function, the study of its fixed points is straightforward.

When $\sigma_r < \sqrt{2}$, h_1 has a unique fixed point a given by

$$a = \frac{\sigma_i^2}{2-\sigma_r^2}. \tag{36}$$

Since $h_1(x) \geq x$ for $x \leq a$, G_{eq} is an increasing sequence, and therefore converges to a.

We define the function h_2 on $[0,a]$ by

$$h_2^g : x \mapsto \frac{1}{2\pi}\left(\sigma_r^2 x+\sigma_i^2\right)\arccos\left(-\frac{\sigma_r^2 x+\sigma_i^2}{\sigma_r^2 g+\sigma_i^2}\right) + \frac{1}{2\pi}\sqrt{\left(\sigma_r^2 g+\sigma_i^2\right)^2-\left(\sigma_r^2 x+\sigma_i^2\right)^2}. \tag{37}$$

We have $G_{\text{neq}}^{(t+1)} = h_2^{G_{\text{eq}}^{(t)}}\left(G_{\text{neq}}^{(t)}\right)$. As G_{eq} converges to a, we will study the sequence g_{neq} defined by $g_{\text{neq}}^{(0)} = 0$ and $g_{\text{neq}}^{(t+1)} = h_2^a\left(g_{\text{neq}}^{(t)}\right)$.

First of all, a is a fixed point of h_2^a. We then compute the first derivative:

$$(h_2^a)'(x) = \frac{1}{2\pi}\sigma_r^2\arccos\left(-\frac{\sigma_r^2 x+\sigma_i^2}{\sigma_r^2 a+\sigma_i^2}\right). \tag{38}$$

In particular, $(h_2^a)'(a) = \sigma_r^2/2 < 1$.

Moreover the second derivative is

$$(h_2^a)''(x) = \frac{\sigma_r^4}{2\pi\sqrt{(\sigma_r^2 a+\sigma_i^2)^2-(\sigma_r^2 x+\sigma_i^2)^2}} > 0 \tag{39}$$

for all $x \in [0,a)$. Thanks to these two inequalities, a is the unique fixed point of h_2^a in $[0,a]$, and $h_2^a(x) > x$ for $x \in [0,a)$. g_{neq} is an increasing sequence bounded above, thus converges towards a, the fixed point of h_2^a.