Criticality is all you need

How to initialize recurrent neural networks?

Tommaso Fioratti

Contents

1 Introduction

The aim of this exploratory work is to gain a clearer understanding of how recurrent neural networks (RNNs) should be initialized, by connecting theoretical insights from the physics community with the practical needs of machine learning. It is well known that deep learning models become more expressive as they grow in depth [1], but they are also harder to train: information carried by the input may fail to propagate across many layers or time steps, leading to vanishing or exploding signals.

Prior work [2, 1, 3] has shown that untrained random networks exhibit an order–chaos phase transition at a critical value of the gain, i.e. the variance with which weights are initialized. At this phase transition, also called *edge of chaos*, information propagation is maximized and learning is believed to be most effective [3]. Building on these ideas, we propose a random matrix theory argument [4] to determine the critical gain in a broad class of RNN architectures.

We then test these theoretical considerations in a reservoir computing setup. In this paradigm, a dynamical system, called the reservoir, is driven by an input signal and generates a rich, nonlinear embedding of the input history in a high-dimensional state space [5]. A linear *readout* is then trained on top of this dynamical system to produce the desired output. Crucially, the dynamics of the reservoir itself are left unchanged during learning: only the parameters of the readout are adapted.

In our setting, the reservoir is an RNN initialized according to our random matrix argument, and its recurrent weights are sampled once at initialization and then kept fixed. A linear readout is trained to predict a chaotic time series, the Mackey–Glass system [6]. In this way, we can isolate the impact of the initialization scheme and show that the performance of the reservoir peaks in the critical, edge-of-chaos regime predicted by our theory.

2 Sompolinsky Model

The seminal work of Sompolinsky et al. [2] introduced a simple model of a recurrent neural network with random weights, which exhibits a phase transition from ordered to chaotic dynamics as the gain parameter is varied. The model consists of N neurons $h_i(t)$, updated according to the following dynamics:

$$\tau \frac{dh_i(t)}{dt} = -h_i(t) + \sum_{j=1}^{N} U_{ij} \tanh(h_j(t)),$$

where U_{ij} are the synaptic weights, drawn independently from a gaussian distribution with zero mean and variance g^2/N, and g is the gain parameter controlling the strength of the recurrent connections. The model has been extensively studied as it provides an exact analytical description of the phase transition in the limit of large N and can be used to study the dynamics of recurrent machine learning models.
Since our focus is more on the latter, we consider the discrete-time version of the model:

$$h_i(t+1) = \sum_{j=1}^{N} U_{ij} \tanh(h_j(t)), \tag{1}$$

which can be obtained by Euler discretization of the continuous time dynamics with time step $\Delta t = \tau$ and can be written also in the more familiar form:

$$x(t+1) = \tanh(Ux(t)).$$

The discretized version of the model retains the same phenomenology of the continuous time one, and it is more suitable for our purposes as it resembles more closely the RNN architectures used in practice.
In the following we proceed to sketch a formal derivation of the main results in [2], i.e. the order to chaos phase transition of the system. We use a dynamical mean-field theory approach, in which we track the evolution of the statistics of the fields $h_i(t)$ over time. For expository clarity, however, we present a self-contained derivation based on the assumption that the disorder becomes decorrelated from the states in the thermodynamic limit. This assumption is the key ingredient that can be rigorously justified within the generating functional formalism, and we refer the reader to [7, 8] for the full derivation without any assumption.

2.1 Order to chaos phase transition via DMFT

Let us consider two replicas under the same disorder, i.e. the same weight matrix U, but independent initial conditions $h_i^1(0)$ and $h_i^2(0)$. Let $d(t)$ track the average square distance between the two replicas at time t:

$$d(t) = \frac{1}{N}\|\delta h(t)\|^2 = \frac{1}{N}\sum_{i=1}^{N}(h_i^1(t) - h_i^2(t))^2.$$

As usually made in dynamical systems theory we consider the distance to be infinitesimal, so that we can linearize the dynamics:

$$h_i^1(t+1) - h_i^2(t+1) = \sum_{j=1}^{N} U_{ij}\tanh'(h_j(t))(h_j^1(t) - h_j^2(t)),$$

where we have defined $h_j(t) = (h_j^1(t) + h_j^2(t))/2$ as the average path.
Now we substitute the above expression into the definition of $d(t)$:

$$\begin{aligned} d(t) &= \frac{1}{N}\sum_{i=1}^{N}(h_i^1(t) - h_i^2(t))^2 \\ &= \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{l=1}^{N} U_{ij}\,U_{il}\,\tanh'(h_j(t))\,\tanh'(h_l(t))\,(h_j^1(t) - h_j^2(t))\,(h_l^1(t) - h_l^2(t)). \quad (2) \end{aligned}$$

Using the fact that the weights are iid with zero mean and variance g^2/N and that averaging over the sites is equivalent to averaging over the disorder (self-averaging hypothesis) and that for large enough N the law of the weights is independent from the law of the fields $h_j(t)$, $\delta h_j(t)$, we get that the only terms that survive are those with $j = l$:

$$d(t) = g^2\frac{1}{N}\sum_{j=1}^{N}\tanh'(h_j(t))^2(\delta h_j(t))^2. \quad (3)$$

Now note that there is no reason why h^1 and h^2 should have different statistics for long enough time as they are replicas of the same system, thus we assume they are independent and identically distributed random variables with gaussian statistics with zero mean and variance $\sigma^2(t)$.
Given this, it is trivial to show that the covariance between the centre of mass $(h^1+h^2)/2$ and the difference $\delta h = h^1 - h^2$ is zero, thus we can factorize the average over the sites and obtain:

$$d(t) \;=\; g^2\underbrace{\left\langle \tanh'\big(h_j(t)\big)^2\right\rangle}_{f(\sigma^2(t))}\, d(t-1)\,,$$

$$f(\sigma^2(t)) = \int\frac{dz}{\sqrt{2\pi}}\,e^{-z^2/2}\,\tanh'\big(\sqrt{\sigma^2(t)}\,z\big)^2 = \int\frac{dz}{\sqrt{2\pi}}\,e^{-z^2/2}\,\mathrm{sech}^4\big(\sqrt{\sigma^2(t)}\,z\big). \quad (4)$$

so that if we know how the variance of the fields evolves in time we can track the evolution of the distance between two replicas.

To close the system of equations we need to find an expression for $\sigma^2(t)$. From (1) we have:

$$\sigma^2(t+1) = \frac{1}{N}\sum_{i=1}^{N} h_i(t+1)^2 = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1,k=1}^{N} U_{ij}U_{ik}\tanh(h_j(t))\tanh(h_k(t)),$$

and again by independence of the weights and self-averaging we get in the thermodynamic limit:

$$\sigma^2(t+1) = g^2 \int \frac{dz}{\sqrt{2\pi}} e^{-z^2/2}\tanh(\sqrt{\sigma^2(t)}z)^2. \tag{5}$$

Supposing now that the variance of the trajectories reaches a fixed point, so that we are analyzing perturbation only after long enough time, we can set $\sigma^2(t) = \sigma^2(t+1) = \sigma_*^2$ and analyze the stability of $d(t)$:

$$d(t) = g^2 f(\sigma_*^2) d(t-1) \quad \text{with} \quad \sigma_*^2 = g^2 \int \frac{dz}{\sqrt{2\pi}} e^{-z^2/2}\tanh(\sqrt{\sigma_*^2}z)^2, \tag{6}$$

which depends only on the value of the factor $g^2 f(\sigma_*^2)$.

We present in Figures 1 and 2 the numerical solution of the self-consistent equation for the variance fixed point and the corresponding perturbation growth factor as a function of the gain g.

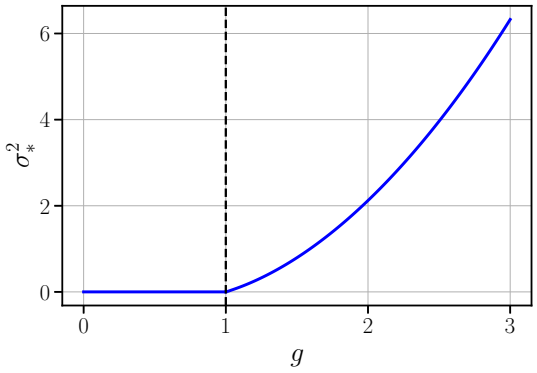

Figure 1: Stationary variance.

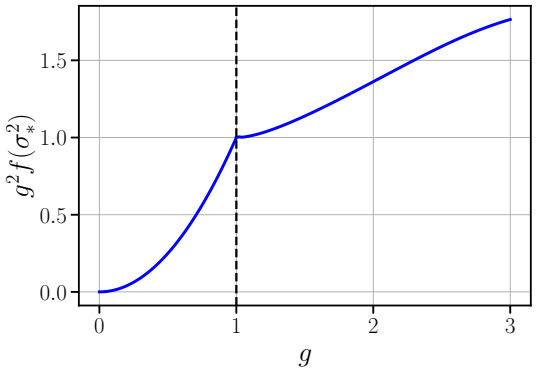

Figure 2: Perturbation growth factor.

Note that zero is always a solution of the self-consistent equation for the variance, but this solution is stable only for $g < 1$. Thus, for $g < 1$ we have $f(0) = 1$ and the factor multiplying $d(t-1)$ is less than one guaranteeing that the perturbation decays (ordered phase). On the other hand, for $g > 1$ the variance fixed point is non-zero and we have that $g^2 f(\sigma_*^2)$ is larger than one, leading to an exponential growth of the distance between the two replicas, i.e. chaos.

2.2 Stability of the fixed point

In the ordered phase when the distance between different initial conditions decays to zero, the system has to converge in a global attractor, which can be shown to be the $h = 0$ fixed point [2]. Thus, as long as this fixed point is stable the systems is maintained in the ordered phase and when it becomes unstable the system enters the chaotic phase, with no other possible dynamics. This is a crucial fact that will be used in the next section to generalize the result to other RNN architectures in which dynamical mean field theory is not easily applicable.

We proceed by showing that the stability analysis of the fixed point $h = 0$ leads to the same critical value $g = 1$. To study the stability of this fixed point we can linearize the dynamics around it:

$$h_i(t+1) \approx \sum_{j=1}^{N} U_{ij} h_j(t),$$

and see that the stability is determined by the spectral radius of the weight matrix U. From random matrix theory [9] we know that in the thermodynamic limit, as the number of neurons grows to infinity, the spectrum of U is uniformly distributed in a disk of radius g in the complex plane.

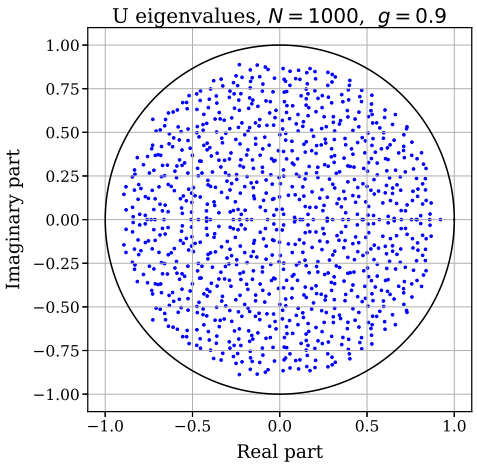

Figure 3: Circular law g = 0.9.

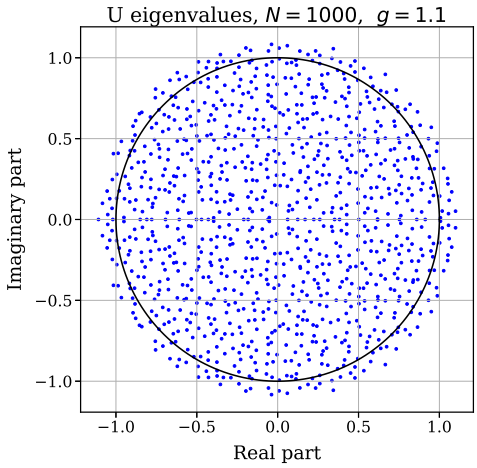

Figure 4: Circular law g = 1.1.

Given this, it is trivial to conclude that for $g < 1$ all eigenvalues have modulus less than one and the fixed point is stable, while for $g > 1$ there are eigenvalues with modulus larger than one and the fixed point is unstable.

3 Multilayer Perceptron

Before moving to gated RNNs, we briefly recall how the same ideas apply to deep feed-forward networks (Multilayer Perceptrons, MLPs), following [1, 3].
We do this for two reasons: first, it shows that the techniques and concepts we have used are very general, second, as we shall show shortly, in the thermodynamic limit, an MLP displays the same phase transition as the discretized Sompolinsky model, despite the different architecture, so that introducing an additional degree of freedom, namely nonzero biases, extends also the previous analysis.
The main structural difference between the two models is that, for an MLP, the "depth" index is the layer index rather than time, and at each step of the propagation the weight matrix changes: each layer has its own independently sampled weight matrix.
We refer to this as annealed disorder, in contrast to the quenched disorder of the recurrent case, where the weight matrix is sampled once and then kept fixed for all times.
We remind the reader that in deducing (3) and (5) we implicitly assumed that, in the thermodynamic limit, the disorder was effectively uncorrelated from the states, while this assumption is already exact independently of N in this annealed setting, where at layer l the matrix $U^{(l)}$ is independent of the preactivations at layer $l-1$.
Thus, all the steps of the derivation hold exactly here, without the need of invoking any additional assumptions or formalism.

3.1 Order to chaos phase transition via DMFT

Let us consider an MLP with L layers, each with N neurons, and tanh activation function. The preactivations at layer l are given by:

$$h_i^{(l)} = \sum_{j=1}^{N} U_{ij}^{(l)} \phi(h_j^{(l-1)}) + b_i^{(l)},$$

where $U_{ij}^{(l)}$ are the weights of layer l, drawn independently from a gaussian distribution with zero mean and variance g^2/N, and $b_i^{(l)}$ are the biases, drawn independently from a gaussian distribution with zero mean and variance σ_b^2.
We consider again two replicas with independent initial conditions $h_i^{1,(0)}$ and $h_i^{2,(0)}$. In this setting, this corresponds to fixing the disorder of the entire MLP and injecting two different inputs in order to study how the distance between their hidden representations evolves with depth.
By following the same computation as before, we obtain the same update for the mean square distance:

$$d(l) = g^2 f(\sigma_*^2) d(l-1),$$

where $f(x) = \langle \tanh'(x) \rangle$ like before and now the variance fixed point depends on the bias:

$$\begin{aligned}\sigma^2(l+1) &= \frac{1}{N}\sum_{i=1}^{N} h_i(l+1)^2 \\ &= \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1,k=1}^{N}\Big(U_{ij}\tanh(h_j(l)) + b_j(l)\Big)\Big(U_{ik}\tanh(h_k(l)) + b_k(l)\Big) \\ &= g^2\int \frac{dz}{\sqrt{2\pi}} e^{-z^2/2}\tanh(\sqrt{\sigma^2(l)}z)^2 + \sigma_b^2\end{aligned}$$

The dynamical mean-field description is the same as in the Sompolinsky model: it yields the same set of self-consistent equations for the mean square distance and the variance (where now we have an additional bias term), which can be numerically solved for any given value of the gain g and the bias variance σ_b^2.
We report in Figure 5 the phase transition curve as a function of the gain and bias standard deviation, obtained by numerically solving the self-consistent equations.

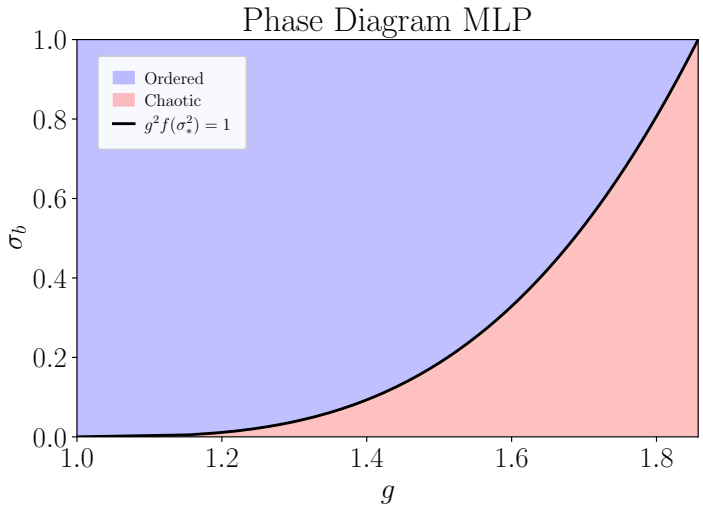

Figure 5: Phase diagram obtained by DMFT analysis of random Multilayer Perceptron.

As shown by (4) the critical gain is controlled by the average slope of the activation function, thus it is intuitive to see that adding a bias increases the critical gain, as it pushes the preactivations away from zero, where the slope of tanh is maximum, towards the saturation regions where the slope is smaller.

3.2 Stability of the fixed point

As before, the same critical value can be obtained by studying the stability of the fixed point, which is not $h = 0$ anymore, but a non-trivial one satisfying:

$$h_i^* = \sum_{j=1}^{N} U_{ij}\,\phi(h_j^*) + b_i.$$

Even if we cannot solve explicitly for h^*, we can study the stability of this fixed point by examining the spectrum of the Jacobian of the dynamics linearized around h^*:

$$J = \text{diag}\left(\phi'(h^*)\right) U, \tag{7}$$

where $\phi = \tanh$ and $\text{diag}(\phi'(h^*))$ denotes the diagonal matrix with entries $\phi'(h_i^*)$. To study the eigenvalues of J, we introduce hereafter the general criterion that will be used also for gated RNNs in the next section.
Let M, L, R be three $N \times N$ deterministic[1] matrices and let U be a random matrix with iid entries with zero mean and variance g^2/N. We consider the matrix:

$$J = M + LUR, \tag{8}$$

then [4] shows that, under some technical assumptions which are always satisfied in our cases, the boundary of the spectrum of J in the thermodynamic limit is given by the set of complex numbers z satisfying

$$g^2 \sum_{i=1}^{N} \sigma_i^{-2} = 1, \tag{9}$$

where σ_i are the singular values of $L^{-1}(zI - M)R^{-1}$.
For the Jacobian of the MLP we have $M = 0$, $L = \text{diag}(\phi'(h^*))$, $R = I$ so that the equation for the border of the spectrum becomes:

$$g^2 \frac{1}{N} \sum_{i=1}^{N} \frac{\phi'(h_i^*)^2}{|z|^2} = 1$$

and the smallest value of the gain for which there exists an eigenvalue with unitary modulus is given by:

$$g_c^2 \langle \phi'(h_i^*)^2 \rangle = 1.$$

Noting that in the thermodynamic limit the distribution of h_i^* is gaussian with zero mean and variance σ_*^2 satisfying the self-consistent equation, we have recovered the same condition obtained from the dynamical mean field theory.

[1] In our case the matrices are random, but since they are independent and have finite moments, we expect the empirical spectral distribution to be self-averaging in the limit $N \to \infty$.

4 Gated RNNs

Now we aim to extend the previous analysis to more complex RNN architectures, in particular gated RNNs such as LSTMs [10] and GRUs [11]. Gated RNNS have been shown to be much more effective than vanilla RNNs in practical applications, as they are able to better capture long-term dependencies in sequential data [12].
However, their dynamics are more complex and a dynamical mean field analysis is not straightforward as the h is not guaranteed in general to have a gaussian distribution.
Thus, as hinted in the previous sections, we rely on the stability analysis of the fixed point to determine the critical gain, even if this is not a global property and it is not a priori linked with chaos, we extend the random matrix argument in this setting and then verify numerically if the system turns indeed in a chaotic attractor for that specific critical gain.
The argument we make is general and can be applied to a broad class of RNN architectures, as long as they can be written in the following general form:

$$h_{t+1,i} = A_{t,i} \left((1 - \alpha_{t,i})\, h_{t,i} + \alpha_{t,i}\, \phi\left(g \sum_j U_{ij}\, o_{t,j}\, \Psi(h_{t,j}) \right) \right). \tag{10}$$

In this formulation:

- $A_{t,i}$ is an amplitude factor modulating the updated state.
- $\alpha_{t,i}$ is a Kauffman update rate controlling the interpolation between the previous hidden state $h_{t,i}$ and the newly computed activation.
- g is the gain controlling the strength of recurrent interactions.
- U_{ij} denotes the recurrent weight matrix, where the entries are iid mean zero gaussian and variance $1/N$.
- $o_{t,j}$ represents the modulation applied to the presynaptic unit.
- $\phi(\cdot)$ is the nonlinearity applied to the aggregated input.
- $\Psi(\cdot)$ is the nonlinearity applied to the internal state $h_{t,j}$.

This generic formulation encompasses several well-known recurrent architectures as special cases[2] and ensures that the system always admits a fixed point at $h = 0$. This property is important for two reasons: first, when using the RNN to predict returns, a zero input at steady state should not induce the model to take any position and the fixed point $h = 0$ enforces this neutrality. Second, the Jacobian at this fixed point takes

[2]With the bias in the candidate state set to zero.

a particularly simple form (8) where the matrices M, L, and R are diagonal and mutually independent. This guarantees that the empirical spectral distribution is likely to be self-averaging in the thermodynamic limit.
In contrast, a non-trivial fixed point $h^* \neq 0$ would generate a sum of dependent random matrices of the type (7), which no longer decomposes in the same way and therefore cannot be reduced in a straightforward way to the [4] setting.

Table 1: Specializations of the unified recurrent update equation.

Architecture	$A_{t,i}$	$\alpha_{t,i}$	$o_{t,i}$	Ψ
RNN	1	1	1	Id
LSTM	$f_{t,i} + i_{t,i}$	$\dfrac{i_{t,i}}{f_{t,i} + i_{t,i}}$	$o_{t,i}$	tanh
GRU	1	$z_{t,i}$	$r_{t,i}$	Id

4.1 Critical gain via linear stability analysis

As shown before, we proceed by studying the stability of the fixed point $h = 0$ by linearizing the dynamics of (10). We obtain the following update:

$$h_{t+1,i} \approx A_i^* \left((1 - \alpha_i^*) h_{t,i} + \alpha_i^* \left(g \sum_j U_{ij} o_j^*(h_{t,j}) \right) \right)$$

where we have defined $A_i^* = A_{t,i}|_{h=0}$, $\alpha_i^* = \alpha_{t,i}|_{h=0}$, and $o_j^* = o_{t,j}|_{h=0}$.
The Jacobian of the dynamics at the fixed point is thus given by:

$$J = D_A \big[(I - D_\alpha) + D_\alpha \, g U D_o \big] \tag{11}$$

where $D_A = \mathrm{diag}(A_i^*)$, $D_\alpha = \mathrm{diag}(\alpha_i^*)$, and $D_o = \mathrm{diag}(o_i^*)$.
Using again the general criterion from [4], we find that the boundary of the spectrum of J is given by the set of complex numbers z satisfying:

$$g^2 \frac{1}{N} \sum_{i=1}^{N} \sigma_i^{-2} = 1 \tag{12}$$

where σ_i are the singular values of

$$D_\alpha^{-1} \, D_A^{-1} (zI - D_A(I - D_\alpha)) D_o^{-1}.$$

Now for easeness of notation we define:

$$M = D_A(1 - D_\alpha), \quad L = D_A D_\alpha, \quad R = D_o$$

so that (12) becomes:

$$g^2 \frac{1}{N} \sum_{i=1}^{N} \frac{L_{ii}^2 R_{ii}^2}{|z - M_{ii}|^2} = 1. \tag{13}$$

where we used the fact that all the matrices are diagonal and invertible, thus the singular values of the product are the products of the singular values, and the inverse of a diagonal matrix is diagonal with entries given by the inverses of the original entries.
Now, for concreteness, let us specialize this result to LSTMs and GRUs, where we have

$$L, M, R = \mathrm{diag}(\sigma(b)) \quad \text{where } \sigma(x) = \frac{1}{1 + e^{-x}}.$$

This means that all the matrices are positive definite with entries in $(0, 1)$.
Given this, it is easy to see that the the smallest value of the gain for which there exists an eigenvalue with unitary modulus is obtained by setting $z = 1$ in (13), so that by simple algebraic manipulations we can find:

$$g_c = \left(\frac{1}{N} \sum_{i=1}^{N} \frac{L_{ii}^2 R_{ii}^2}{(1 - M_{ii})^2} \right)^{-1/2}. \tag{14}$$

Thus, we have obtained a closed-form expression for the critical gain of gated RNNs in terms of the bias design choices, which control the entries of the diagonal matrices L, M, and R. Even if in principle we have to sample the biases to compute the critical gain by this formula, in practice the result is the same for any realization of the biases in the thermodynamic limit and for some cases it can be computed analytically by taking the expectation over the bias distribution. In the cases where it is not possible, sampling and computing the formula is still a very light numerical task.

4.2 Benettin Algorithm

In the previous section we derived an analytical prediction for the critical gain at which the zero fixed point becomes unstable. We now verify numerically whether this critical value indeed marks the onset of chaos in the dynamics. To do so, we compute the maximal lyapunov exponent

$$\lambda_{max} = \lim_{t \to \infty} \frac{1}{t} \log \frac{\|\delta h(t)\|}{\|\delta h(0)\|},$$

as a function of the gain g using the Benettin algorithm [13]. The maximal Lyapunov exponent quantifies the average exponential rate of divergence of infinitesimally close trajectories: if $\lambda_{max} > 0$ the system is chaotic, while if $\lambda_{max} < 0$ the system is ordered. Thus, the critical gain for the onset of chaos is the value of g such that $\lambda_{max} = 0$.
We consider the discrete-time dynamics $h_{t+1} = F(h_t)$ in Eq. (10) and its tangent evolution given by the Jacobian $J(h_t)$. Starting from a random initial condition h_0 and a

random unit vector u_0, we iterate in parallel

$$h_{t+1} = F(h_t), \qquad u_{t+1} = J(h_t)\, u_t.$$

We warm up the system for a time T_{trans}. Then, at regime we compute $\tilde{u}_{t+1} = J(h_t)u_t$, accumulate $\log \|\tilde{u}_{t+1}\|$ and renormalize $u_{t+1} = \tilde{u}_{t+1}/\|\tilde{u}_{t+1}\|$. After T_{sim} iterations, the finite-time estimate of the maximal Lyapunov exponent is

$$\lambda = \frac{1}{T_{\text{sim}} - T_{\text{trans}}} \sum_{t=T_{\text{trans}}}^{T_{\text{sim}}-1} \log \|\tilde{u}_{t+1}\|.$$

Then λ_{max} is obtained by averaging this estimator over S independent realizations.

Algorithm 1 Benettin algorithm for the maximal Lyapunov exponent λ_{max}

Require: number of samples S; total iterations T_{sim}; transient length T_{trans}

for $s = 1, \ldots, S$ **do**
  Draw random disorder and initial state h_0
  Choose random unit vector u_0
  $h \leftarrow h_0,\ u \leftarrow u_0,\ r \leftarrow 0$
  for $t = 0, \ldots, T_{\text{sim}} - 1$ **do**
    $h \leftarrow F(h)$
    $\tilde{u} \leftarrow J(h)\, u$
    if $t \geq T_{\text{trans}}$ **then**
      $r \leftarrow r + \log \|\tilde{u}\|$
    end if
    $u \leftarrow \tilde{u}/\|\tilde{u}\|$
  end for
  $\lambda_s \leftarrow r/(T_{\text{sim}} - T_{\text{trans}})$
end for
return $\lambda_{max} = \frac{1}{S} \sum_{s=1}^{S} \lambda_s$

The Benettin algorithm provides a numerically stable way to estimate the maximal Lyapunov exponent by evolving an infinitesimal perturbation under the tangent dynamics and periodically renormalizing its norm, so as to remain in the linear regime and avoid saturation effects that would affect a naive finite-difference estimate based on the separation of two full trajectories.

Using the algorithm, we now test whether the critical gain associated with the instability of the fixed point also predicts the onset of chaos. In the zero bias case, plugging $\sigma(0) = \frac{1}{2}$ into Eq. (14) directly gives $g_c = 2$.

Figure 6 hereafter shows the resulting maximal Lyapunov exponent as a function of the gain g.

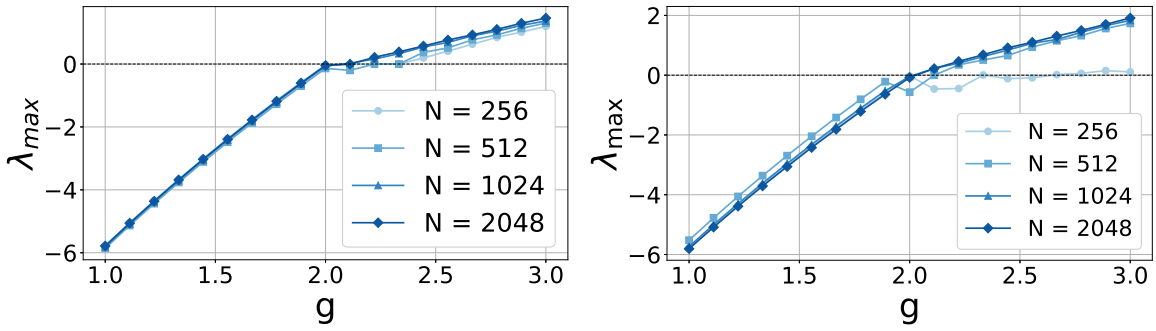

Figure 6: Maximum Lyapunov exponent estimated with the Benettin algorithm as a function of the gain g for LSTM (left) and GRU (right) with zero biases with different sizes N.

As we can see from the figure, the maximal Lyapunov exponent crosses zero exactly at the predicted critical gain $g_c = 2$, confirming that the instability of the fixed point indeed marks the onset of chaos in the dynamics.

4.3 Phase Transitions

Now we aim to extend this analysis to the case of non-zero biases, where the critical gain depends on the bias design choices.
We consider here two different bias initialization schemes: a gaussian initialization where all biases are drawn independently from a Gaussian distribution with zero mean and standard deviation σ_b, and a popular initialization scheme called the chrono initialization [14]. These choices are arbitrary and we stress that the design tool we provide can be applied to any bias initialization that keeps the fixed point at zero.
For each bias initialization scheme, we compute the critical gain using Eq. (14) and verify numerically whether this gain predicts the onset of chaos using the Benettin algorithm. The results are reported in Figures 8 and 7, showing a perfect agreement between the theoretical prediction and the numerical estimate of the critical gain for both LSTM and GRU architectures.

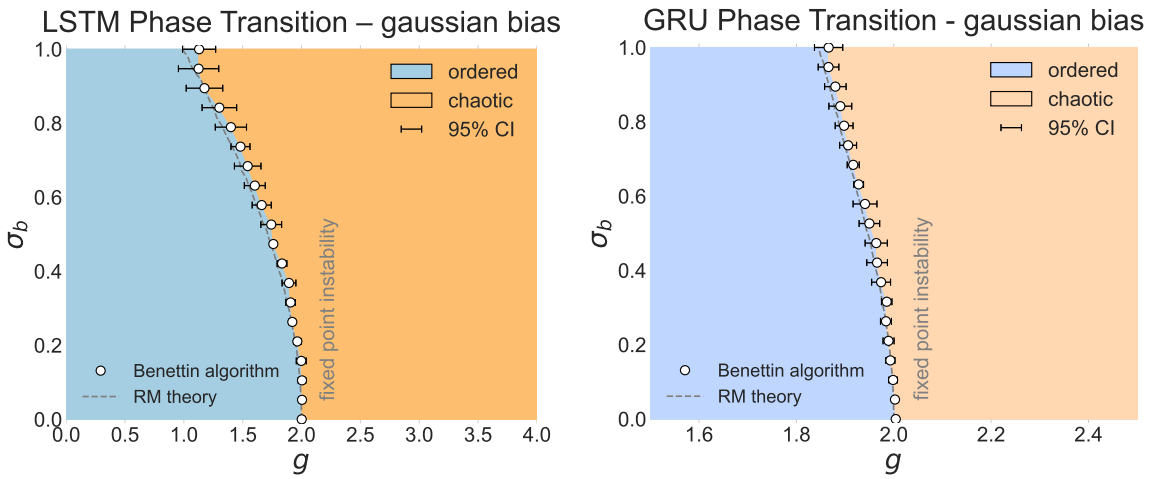

Figure 7: Phase transition curves for LSTM and GRU as a function of the Gaussian bias standard deviation σ_b. The dashed line indicates the theoretical prediction from Eq. (14).

Interestingly and contrary to the MLP case, increasing the bias variance decreases the critical gain.

The effect of the bias variance on the critical gain can be clarified by examining the Jacobian in (11). Consider for simplicity a GRU architecture, where according to Table 1, we have, that the Jacobian reduces to

$$J = I - D_\alpha + D_\alpha \, g \, U \, D_o.$$

The point at which the spectral boundary of J intersects the unit circle depends only on the spectral radius of the matrix $g\,U\,D_o$. This is evident from (14), since $M = I - D_\alpha$ and $L = D_\alpha$, the terms $(1 - M_{ii})^2$ and L_{ii}^2 cancel out. Then, since the critical gain is inversely proportional to the variance of the entries of D_o, and these entries are given by $\sigma(b_i)$ (σ here refers to the sigmoid function), increasing the bias variance decreases the critical gain.

Note that this behaviour is consistent with the intuition from the Sompolinsky model. The presence of the multiplicative diagonal term D_o effectively rescales each column of the weight matrix by the corresponding factor $\sigma(b_j)$, which is an independent site-dependent random variable. If the original weights U_{ij} were i.i.d. with zero mean and variance g^2/N, the renormalized weights $U_{ij}\,\sigma(b_j)$ acquire a new variance given by:

$$\langle (U_{ij}\,\sigma(b_j))^2 \rangle = \langle U_{ij}^2 \rangle \, \langle \sigma(b_j)^2 \rangle = \frac{g^2}{N} \, \langle \sigma(b)^2 \rangle.$$

where the average is over the bias distribution:

$$\langle \sigma(b)^2 \rangle = \int_{-\infty}^{+\infty} \frac{1}{(1+e^{-b})^2} \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp\left(-\frac{b^2}{2\sigma_b^2} \right) \, db,$$

which is increasing with σ_b.
Thus, increasing the bias variance increases the variance of the effective renormalized weights, leading to a decrease in the critical gain g.
A similar reasoning applies to the LSTM case, even if the expressions are more involved due to the non simplification of the matrices M, L.
Then we analyzed the chrono initialization, which corresponds to taking in our notation framework:

$$M = I - \mathrm{Diag}\big(1/\tau_i\big), \quad L = \mathrm{Diag}\big(1/\tau_i\big), \quad R = \tfrac{1}{2}I, \quad \tau_i \sim \mathcal{U}(2, T_{\max}).$$

where $T_{\max}$ is a hyperparameter controlling the maximum timescale that the time series we want to forecast should have.
This initialization is motivated by the idea of enforcing the model to have memory timescales that are compatible with the timescales of the input data. The interesting fact for our design tool is that this initialization enforces a special symmetry that keeps the critical gain at a constant value independently of the hyperparameter $T_{\max}$.

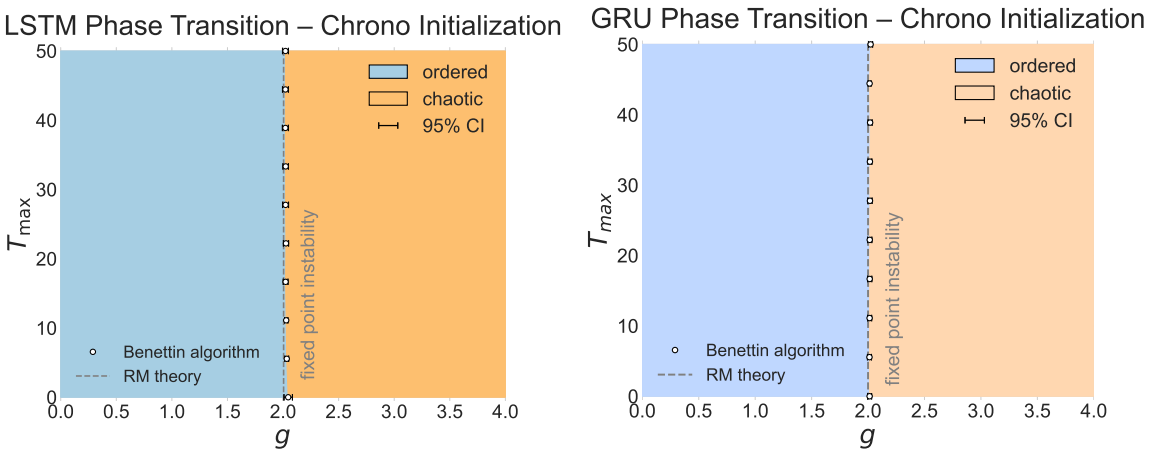

Figure 8: Phase transition curves for LSTM and GRU with chrono initialization. The dashed line indicates the theoretical prediction from Eq. (14).

Note that the symmetry enforced by the chrono initialization in the LSTM is the one that the GRU naturally exhibits for its own architecture $f_t = 1 - i_t$.
As you can check from (13), the matrices that encode the dependence on $T_{\max}$ cancel out in determining the critical gain, yielding $g_c = 2$. Again, we find perfect agreement between the theoretical prediction and the numerical estimate of the onset of chaos, both in Figure 8 for LSTM and GRU architectures, and for any other GRU initialization that modifies only b_z, which produces the same phase transition curve, due to the symmetry discussed above (figure not shown).

5 Reservoir Computing

To empirically validate the theoretical predictions regarding the critical gain in gated RNNs, we employ a reservoir computing framework. In this setup, we initialize a gated recurrent neural network (RNN) with fixed random weights and use it as a dynamical reservoir to process input signals. A linear readout layer is then trained to map the reservoir states to the desired output.

Reservoir computing is at its fundament a kernel regression method, where the reservoir acts as a non-linear feature map that transforms the input data into a sufficiently high-dimensional representation such that the task is linearly separable in the augmented space.

Given this, it is easy to understand that the properties of the reservoir, such as its dynamical regime, will play a role in determining the quality of the feature map and, consequently, the performance of the regression task. The reservoir has to necessarily remember a sufficient amount of information from the input history but also be sufficiently expressive to transform the input into a rich representation, which is exactly what happens at the edge of chaos. Thus, it has been hypothesized and empirically observed that initializing the reservoir at the edge of chaos leads to optimal performance [15, 16]. In this framework, we can thus verify if there is agreement between the optimal performance and the critical gain predicted by our theory.

We consider the task of predicting the Mackey-Glass chaotic time series [6], a standard benchmark in the reservoir computing literature. The Mackey-Glass system is defined by the difference equation:

$$u_(t+1) = (1-\gamma)u(t) + \beta \frac{u(t-\tau)}{1+u(t-\tau)^n}$$

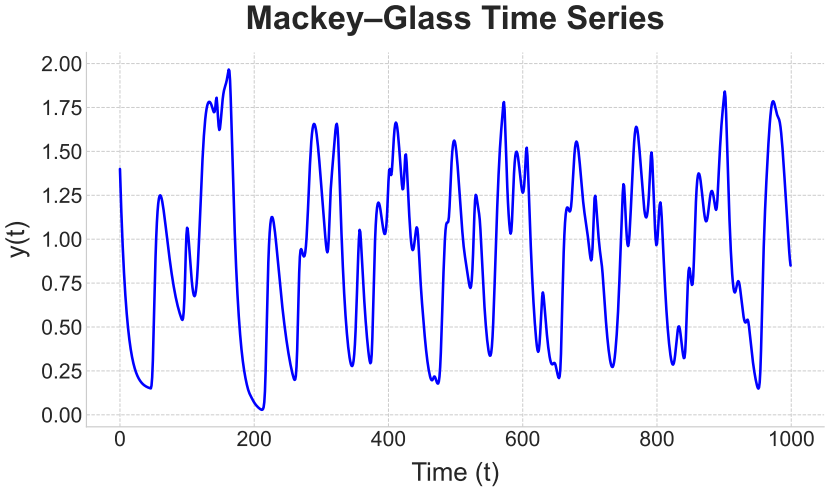

Figure 9: Sample of the Mackey-Glass chaotic time series.

and we consider a set of standard parameters $\beta = 0.2$, $\gamma = 0.1$, $n = 10$, and $\tau = 17$ which yield chaotic dynamics. The choice of the time series is motivated by its use in the literature but we found agreement also on other chaotic time series such as the logistic map (figure not shown).

We first simulated reservoirs using LSTMs with zero bias and studied the training and test loss as a function of the gain g for different reservoir sizes N, to both verify the theoretical prediction and check for finite size effects.

The results reported in Figure 10 show a clear minimum of the mean squared error (MSE) at a gain value that is in very good agreement with the critical gain predicted by the instability of the trivial fixed point.

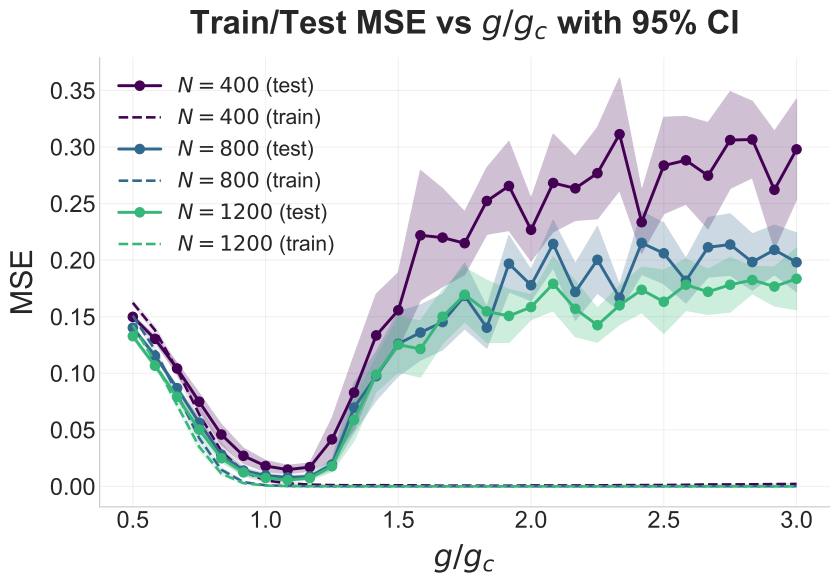

Figure 10: Mean squared error (MSE) of the prediction as a function of the gain for different reservoir sizes N.

The minimum is in fact attained at a gain that is only minimally larger than the theoretical critical value, a small systematic shift that can be attributed to the presence of an external input, which is not accounted for in the theoretical analysis and tends to suppress the chaotic dynamics. Although we did not investigate this effect systematically, we empirically observed that, as the input amplitude is reduced, the prediction accuracy improves and the optimal performance approaches the gain closest to the theoretical critical value; upon further decreasing the input amplitude, the performance starts to degrade (figure not shown).

We also see that the MSE is lower for bigger reservoirs, indicating that for the size considered the regression is not overfitting yet and the bigger dimensionality of the feature space helps in improving the performance.

The last observation we want to make is that the training loss does not show a minimum

at the critical gain, but has instead a monotonic decreasing behaviour, being very low in the chaotic regime. This is expected, as in the chaotic regime the reservoir is sufficiently expressive to memorize the training set but fails to generalize due to the sensitivity to initial conditions, i.e. the regression is overfitting.
Lastly, we fixed the reservoir size to $N = 2000$ and extended the analysis in the case of a varying bias standard deviation σ_b, where all the biases were drawn to be gaussian like in Figure 7.

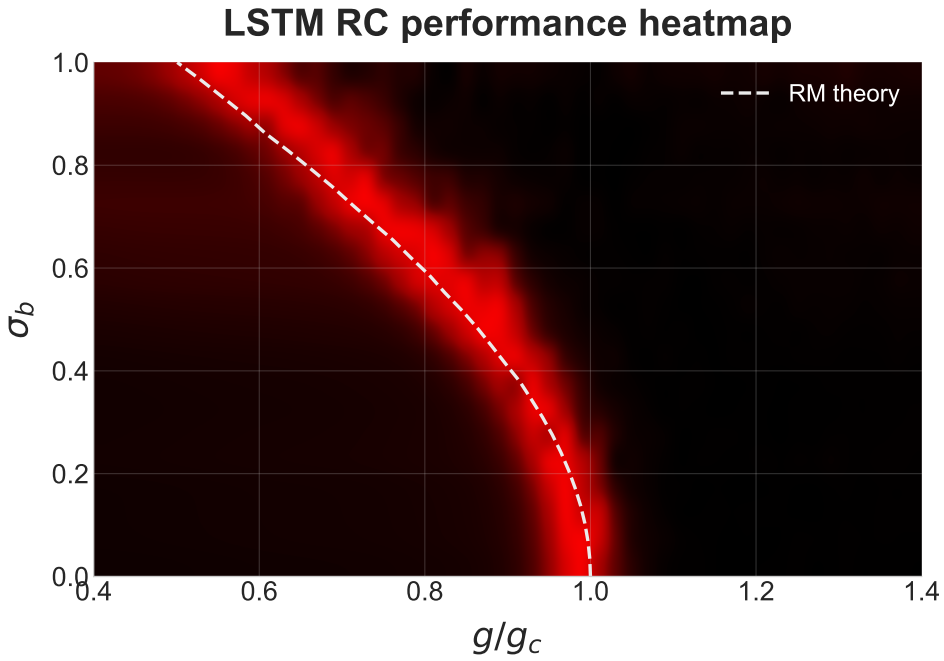

Figure 11: Heatmap of the reservoir computing performance, measured as $1/MSE$ as a function of the gain g and the bias standard deviation σ_b. The dashed white line is the theoretical critical gain from Eq. (14).

The results, reported in Figure 11, show again a clear peak of performance along the theoretical critical line predicted by our analysis, confirming that initializing the reservoir at the edge of chaos leads to optimal performance even in the bias case.

6 Conclusions

In this work, we used insights from Dynamical Mean Field Theory (DMFT) of vanilla recurrent neural networks (Sompolinsky model) and Multilayer Perceptrons to derive a design tool for the critical gain at which more complicated models, i.e. gated RNNs, transition from an ordered to a chaotic phase.
The key idea was to study the linear stability of the trivial fixed point at zero, which below the phase transition acts as a global attractor of the dynamics, and derive a closed-form expression for the critical gain using random matrix theory.
We then verified numerically, using the Benettin algorithm, that this critical gain, associated with the instability of the trivial fixed point, indeed predicts the onset of chaos in the dynamics of GRUs and LSTMs, for different bias initializations.
Building on this, we applied our theoretical findings in a reservoir computing framework and empirically verified that initializing the reservoir at the edge of chaos leads to optimal performance on a chaotic time-series prediction task.
This not only confirms the practical relevance of our analysis and provides a useful guideline for designing gated RNNs in reservoir computing applications, but also extends previous results by considering more complex reservoir architectures than standard echo state networks, thereby moving towards the recent trend of employing diverse systems as reservoirs and tuning them to their optimal working point.
Future work should focus on establishing a rigorous connection between the instability of the trivial fixed point and the onset of chaos in this kind of disordered network, potentially by extending or going beyond the DMFT formalism to better capture the dynamics of complex gated RNNs.

References

[1] Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[2] Haim Sompolinsky, Andrea Crisanti, and Hans-Jürgen Sommers. Chaos in random neural networks. *Physical Review Letters*, 61(3):259–262, 1988.

[3] Samuel S. Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. In *International Conference on Learning Representations*, 2017.

[4] Yashar Ahmadian, Francesco Fumarola, and Kenneth D. Miller. Properties of networks with partially structured and partially random connectivity. *Physical Review E*, 91(1):012820, 2015.

[5] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002.

[6] Michael C. Mackey and Leon Glass. Oscillation and chaos in physiological control systems. *Science*, 197(4300):287–289, 1977.

[7] Jannis Schuecker, Sven Goedeke, and Moritz Helias. Optimal sequence memory in driven random networks. *Physical Review X*, 8(4):041029, 2018.

[8] Lutz Molgedey, Johannes Schuchhardt, and Hermann G. Schuster. Suppressing chaos in neural networks by noise. *Physical Review Letters*, 69(26):3717–3719, 1992.

[9] Terence Tao and Van Vu. Random matrices: Universality of local spectral statistics of non-hermitian matrices. *Annals of Probability*, 38(5):2023–2065, 2010. See also their proof of the circular law.

[10] Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. In *Neural Computation*, 2000. Introduces the forget gate; modern c_t, h_t, i_t, f_t, o_t notation.

[11] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *EMNLP*, 2014. Introduces the Gated Recurrent Unit (GRU).

[12] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv:1412.3555*, 2014. Compares LSTM, GRU, and vanilla RNNs on long-range sequence tasks.

[13] Giancarlo Benettin, Luigi Galgani, Antonio Giorgilli, and Jean-Marie Strelcyn. Lyapunov characteristic exponents for smooth dynamical systems and for hamiltonian systems; a method for computing all of them. part 1: Theory. *Meccanica*, 15:9–20, 1980.

[14] Julien Tallec and Yann Ollivier. Can recurrent neural networks warp time? In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018. arXiv:1804.11188.

[15] Nils Bertschinger and Thomas Natschläger. Real-time computation at the edge of chaos in recurrent neural networks. *Neural Computation*, 16(7):1413–1436, 2004.

[16] Joschka Boedecker, Oliver Obst, Joseph T. Lizier, N. Michael Mayer, and Minoru Asada. Information processing in echo state networks at the edge of chaos. *Theory in Biosciences*, 131(3):205–213, 2012.