ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

Local Lyapunov Exponents of Deep RNN

Claudio Gallicchio[1] and Alessio Micheli[1] and Luca Silvestri

1 - Department of Computer Science, University of Pisa
Largo Bruno Pontecorvo 3 - 56127 Pisa, Italy

Abstract. The study of deep Recurrent Neural Network (RNN) models represents a research topic of increasing interest. In this paper we investigate layered recurrent architectures under a dynamical system point of view, focusing on characterizing the fundamental aspect of stability. To this end we provide a framework that allows the analysis of deepRNN dynamical regimes through the study of the maximum among the local Lyapunov exponents. Applied to the case of Reservoir Computing networks, our investigation also provides insights on the true merits of layering in RNN architectures, effectively showing how increasing the number of layers eventually results in progressively less stable global dynamics.

1 Introduction

Recently, we are witnessing a growing interest in the extension of deep learning methodologies to the field of Recurrent Neural Networks (RNNs) [1, 2]. In this concern, the introduction of hierarchically organized deepRNN architectures [3, 4] has opened the way to the possibility of developing a temporal data representation at different levels of abstraction, thereby allowing to naturally address tasks on time-series featured by a multiple time-scales organization. However, the study of layered RNNs is still in its initial phases, and further research effort is required especially to characterize the properties of the resulting state dynamics as well as to investigate the actual role of layering in this regard.

In their essential nature, recurrent neural models implement dynamical systems whose trajectories are influenced by initial conditions and by driving input signals. In this context, an aspect of primary importance that is still demanded in literature is represented by the analysis of deepRNNs in terms of stability of networks dynamics guided by an external input. In this paper we address this fundamental problem by providing a theoretical and practical tool that extends the applicability of the study of local Lyapunov exponents [5] to the case of layered recurrent architectures. Although the developed approach can be exploited to investigate system stability at any stage of training, in this paper we experimentally show its application in the significant case represented by untrained networks dynamics under the Reservoir Computing (RC) framework [6]. To this aim, we take as reference model the deep Echo State Network (deepESN) [7, 8], which allows us to investigate the properties of stacked recurrent dynamics in a separate fashion from learning. Considered under this perspective, our analysis also provides insights that shed light on the effective meaning of layering in recurrent neural architectures.

2 Stability of Deep Recurrent Neural Networks

A deepRNN implements an input-driven discrete-time non-linear dynamical system, in which the state dynamics is realized by means of a hierarchical neural network architecture with N_L stacked recurrent layers. At each time step t the state computation proceeds by following a pipeline from the external input to the higher layer. As graphically illustrated in Fig. 1, the first layer receives the external input $\mathbf{u}(t) \in \mathbb{R}^{N_U}$, and each successive layer is fed by the output of the previous one at the same time step.

Focusing only on the state dynamics and assuming, for the sake of simplicity, that the dimension of the state space is the same for every layer, we denote by $\mathbf{x}^{(i)}(t) \in \mathbb{R}^{N_R}$ the state of layer i at time t. Considering leaky integration recurrent units [9, 2], we indicate by $a^{(i)} \in [0,1]$ and $\hat{\mathbf{W}}^{(i)}$ the leaking rate parameter and the recurrent weight matrix of layer i, respectively. Moreover, we use $\mathbf{W}^{(1)}$ to denote the input weight matrix for the first layer, and $\mathbf{W}^{(i)}$ for representing the weight matrix corresponding to the connections from layer $i-1$ to layer i. Based on this notation, the state transition function of the first layer, i.e. $F^{(1)}$, can be computed as follows:

$$\begin{aligned}\mathbf{x}^{(1)}(t) = \;& F^{(1)}(\mathbf{u}(t), \mathbf{x}^{(1)}(t-1)) = \\ & (1-a^{(1)})\mathbf{x}^{(1)}(t-1) + a^{(1)} \tanh(\mathbf{W}^{(1)}\mathbf{u}(t) + \hat{\mathbf{W}}^{(1)}\mathbf{x}^{(1)}(t-1)).\end{aligned} \tag{1}$$

For layer $i > 1$, observing that (recursively) $\mathbf{x}^{(i)}(t)$ is a function of the activations of layers $1, \dots, i$ at step $t-1$ and of the input at step t, we have that the state transition function $F^{(i)}$ can be computed as:

$$\begin{aligned}\mathbf{x}^{(i)}(t) = \;& F^{(i)}(\mathbf{u}(t), \mathbf{x}^{(1)}(t-1), \dots, \mathbf{x}^{(i)}(t-1)) = \\ & (1-a^{(i)})\mathbf{x}^{(i)}(t-1) + a^{(i)} \tanh(\mathbf{W}^{(i)}\mathbf{x}^{(i-1)}(t) + \hat{\mathbf{W}}^{(i)}\mathbf{x}^{(i)}(t-1)) = \\ & (1-a^{(i)})\mathbf{x}^{(i)}(t-1) + a^{(i)} \tanh(\mathbf{W}^{(i)}F^{(i-1)}(\mathbf{u}(t), \mathbf{x}^{(1)}(t-1), \dots, \\ & \qquad\qquad \mathbf{x}^{(i-1)}(t-1)) + \hat{\mathbf{W}}^{(i)}\mathbf{x}^{(i)}(t-1)).\end{aligned} \tag{2}$$

Viewing the deepRNN state evolution as a whole, we can consider the global state space of the network as the product of the state spaces of the N_L layers, and denote by $\mathbf{x}(t) = \big(\mathbf{x}^{(1)}(t), \dots, \mathbf{x}^{(N_L)}(t)\big) \in \mathbb{R}^{N_L \times N_R}$ the global state of the deepRNN at step t. Accordingly, the relation between the global state at two consecutive time steps is given by a state transition function $F = \big(F^{(1)}, \dots, F^{(N_L)}\big)$.

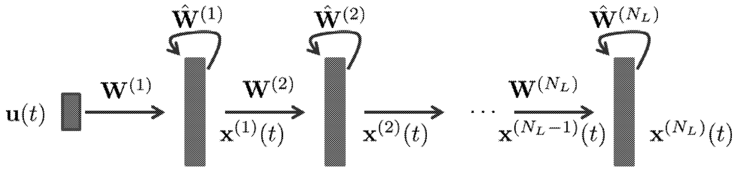

Fig. 1: Architecture of a deepRNN.

The stability of the dynamical system governed by F can be investigated by studying the spectrum of its Lyapunov exponents, which provide a measure of the exponential divergence of trajectories starting in neighboring initial states. The rate of divergence is dominated by the value of the maximum Lyapunov exponent, used as an indicator of the stable/unstable dynamics regime of the system, with values below 0 (resp. above 0) characterizing stable (resp. unstable) dynamics, and a value of 0 denoting a transition condition (towards chaotic behavior) known as critical point [10]. In common practical situations, it is useful to consider the local Lyapunov exponents [5], i.e. local finite-time approximations of the Lyapunov exponents evaluated over a trajectory followed by the system driven by a real input sequence. Accordingly, given an input sequence of length N, the maximum Lyapunov exponent can be approximated by the quantity $\lambda_{max} = \max_k \frac{1}{N}\sum_{t=1}^{N} \log(\mathrm{eig}_k(\mathbf{J}_{F,\mathbf{x}}(t)))$, where $\mathrm{eig}_k(\cdot)$ denotes the module of the k-th eigenvalue of its matrix argument and $\mathbf{J}_{F,\mathbf{x}}(t)$ is the Jacobian of F evaluated at step t. From an operational point of view, the measure provided by λ_{max} allows us to link the stability/instability characterization of the network's behavior to the actual input signal. Taking into account the layer-wise block organization of F, $\mathbf{J}_{F,\mathbf{x}}(t)$ can be written as a block matrix, as follows:

$$\mathbf{J}_{F,\mathbf{x}}(t) = \begin{pmatrix} \mathbf{J}_{F^{(1)},\mathbf{x}^{(1)}}(t) & \dots & \mathbf{J}_{F^{(1)},\mathbf{x}^{(N_L)}}(t) \\ \mathbf{J}_{F^{(2)},\mathbf{x}^{(1)}}(t) & \dots & \mathbf{J}_{F^{(2)},\mathbf{x}^{(N_L)}}(t) \\ \vdots & \ddots & \vdots \\ \mathbf{J}_{F^{(N_L)},\mathbf{x}^{(1)}}(t) & \dots & \mathbf{J}_{F^{(N_L)},\mathbf{x}^{(N_L)}}(t) \end{pmatrix} \tag{3}$$

where for $i, j = 1, \dots, N_L$, $\mathbf{J}_{F^{(i)},\mathbf{x}^{(j)}}(t)$ is the partial derivative of the state transition function of the i-th layer with respect to the state of the j-th layer at step t. Considering the hierarchical state computation process carried out by the deepRNN, we can notice that the Jacobian in eq. 3 has a lower-triangular block matrix structure, as the state of any layer does not depend on the states of higher layers in the stack, i.e. $\mathbf{J}_{F^{(i)},\mathbf{x}^{(j)}}(t) = \mathbf{0}$ for any $j > i$. Accordingly, the eigenvalues of $\mathbf{J}_{F,\mathbf{x}}(t)$ are given by the set of eigenvalues of the matrices on its block diagonal, i.e. $\mathbf{J}_{F^{(i)},\mathbf{x}^{(i)}}(t)$, for $i = 1, \dots, N_L$. Thereby we can derive the following formula for the computation of λ_{max} of a deepRNN:

$$\lambda_{max} = \max_{i,k} \frac{1}{N} \sum_{t=1}^{N} \log\left(\mathrm{eig}_k\left((1 - a^{(i)})\mathbf{I} + a^{(i)}\mathbf{D}^{(i)}(t)\hat{\mathbf{W}}^{(i)}\right)\right) \tag{4}$$

with $\mathbf{D}^{(i)}(t)$ denoting the diagonal matrix whose non-zero elements are the elements of the vector $\tanh(\mathbf{W}^{(i)} F^{(i-1)}(\mathbf{u}(t), \mathbf{x}^{(1)}(t-1), \dots, \mathbf{x}^{(i-1)}(t-1)) + \hat{\mathbf{W}}^{(i)}\mathbf{x}^{(i)}(t-1))$. It is worth noticing that when the network architecture contains only one layer, i.e. if $N_L = 1$, the formula for the computation of λ_{max} in eq. 4 reduces to the case of standard shallow RNNs (e.g. [11, 12]). Moreover, a closer inspection of eq. 4 reveals that the value of λ_{max} is a monotonic non-decreasing function of the number of the layers. This essentially means that adding layers to a deepRNN architecture has the intrinsic effect of driving the network's dynamics towards a less (or at most equally) stable regime.

3 Numerical Simulations

In this Section we practically demonstrate the stability analysis of stacked recurrent dynamics developed in Section 2 through an example involving untrained deepRNN architectures. Specifically, basing on the RC paradigm [6] and in particular on the standard (shallow) ESN [13], we consider the deepESN model introduced in [7, 8]. In a deepESN, the parameters of the state update equations $F^{(1)}, \dots, F^{(N_L)}$ are left untrained after initialization. In particular, for every $i = 1, \dots, N_L$, the weights in $\mathbf{W}^{(i)}$ and $\hat{\mathbf{W}}^{(i)}$ are randomly chosen from a uniform distribution over $[-1, 1]$. After that, the values in $\hat{\mathbf{W}}^{(i)}$ are re-scaled to meet a desired spectral radius[1] value, denoted by $\rho^{(i)}$. In our experimental setting we used the same values of the leaking rate and spectral radius parameters for all the layers, i.e. for all $i = 1, \dots, N_L$ we set $a^{(i)} = a$ and $\rho^{(i)} = \rho$, keeping fixed $a = 1$ in all the experiments, while varying the value of ρ in $[0.5, 1.5]$. We considered deep architectures with a number of layers N_L progressively increasing from 1 to 10, and where each layer contained 10 recurrent units (i.e. $N_R = 10$). For every hyper-parametrization, we independently generated 100 network guesses (with different random seeds), and averaged the results over such guesses. As driving input signal we considered a time-series of length 1000, where the input at each time step was a 10-dimensional vector (i.e. $N_U = 10$) whose elements are individually drawn from a uniform distribution in $[-0.5, 0.5]$.

The results of the λ_{max} computation under the considered settings are illustrated in Fig. 2, where progressively lighter colors correspond to higher values of λ_{max}, i.e. to progressively less stable networks dynamics. As it can be seen,

[1] i.e. the maximum of its eigenvalues in modulus, which leads to a control of stability without taking into account the input (as in traditional RC literature).

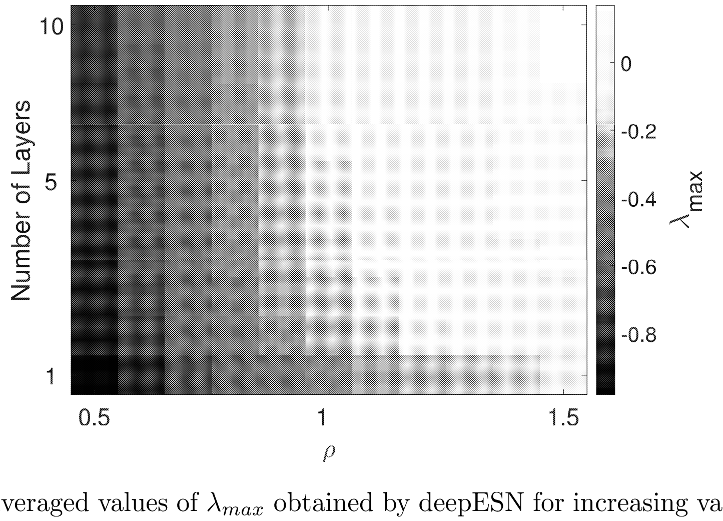

Fig. 2: Averaged values of λ_{max} obtained by deepESN for increasing values of ρ and number of layers.

higher values of the scaling parameter ρ result in higher values of λ_{max} (as observed also in shallow ESNs e.g. in [11]). Moreover, and more importantly, for every value of ρ the value of λ_{max} increases for increasing number of layers in the deep recurrent architecture, eventually switching from stability to a chaotic behavior in correspondence of the higher values of ρ and of the number of layers.

A comparison between the values of λ_{max} of deepESN (with 10 units in each layer) and of standard (shallow) ESN under same hyper-parametrization settings (with $\rho = 1.0$) and for increasing number of recurrent units is reported in Fig. 3. Results clearly point out that organizing the same number of recurrent units into a layered architecture inherently and systematically leads (even prior to learning) to overall network's dynamics characterized by less stable regimes.

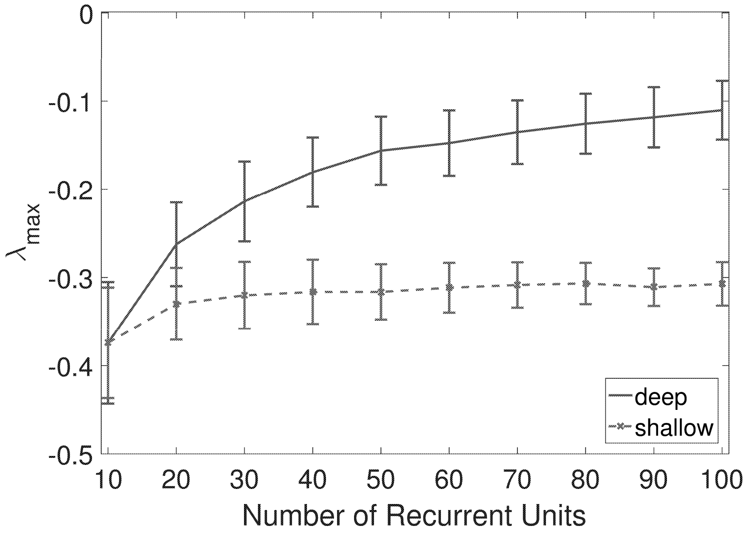

Fig. 3: Averaged values (and standard deviations) of λ_{max} obtained by deepESN and shallow ESN for increasing number of recurrent units.

These findings, on the one hand, suggest us to use caution during the design of a deep recurrent network, as the increase in the number of layers could result in an (undesired) unstable network dynamics. On the other hand, they also explain the potentiality of layered recurrent models in outperforming shallow networks with the same number of recurrent units [7, 8] in tasks on which recurrent models brought to the limit of stability have shown performance maximization, such as the short-term memory capacity as discussed in [10].

4 Conclusions

In this paper we have addressed the problem of analyzing the stability of deep recurrent neural models under a dynamical system perspective. In particular, we have provided a mathematical tool that extends the applicability of the study of local Lyapunov exponents to the case of layered state dynamics evolving in a hierarchical fashion. In the stability analysis, such a tool allows us to practically take into account also the actual input signal influencing the network's behavior.

Our investigation also pointed out interesting insights on the intrinsic effect of layering in RNNs. Specifically, by applying the developed tool to the case of deep RC networks, we have shown that increasing the number of layers in a recurrent architecture has the inherent ability to eventually drive the resulting network's dynamics towards a less stable dynamical regime. Moreover, the same number of recurrent units showed a less stable dynamical behavior when organized in a layered network than in fully-connected shallow cases.

Overall, the approach developed in this paper would contribute to better understand and characterize the properties of state dynamics developed by deep recurrent networks. At the same time, we believe that the proposed methodology will serve as a fruitful base for further developments aimed at guiding the design of deep recurrent networks.

References

[1] P. Angelov and A. Sperduti. Challenges in deep learning. In *Proc. of the 24th European Symposium on Artificial Neural Networks (ESANN)*, pages 489–495. i6doc.com, 2016.

[2] I. Goodfellow, Y. Bengio, and A. Courville. Deep learning. Book in preparation for MIT Press, http://www.deeplearningbook.org, 2016.

[3] M. Hermans and B. Schrauwen. Training and analysing deep recurrent neural networks. In *NIPS*, pages 190–198, 2013.

[4] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio. How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026v5*, 2014.

[5] B.A. Bailey. Local lyapunov exponents: predictability depends on where you are. *Nonlinear Dynamics and Economics, Cambridge University Press*, pages 345–359, 1996.

[6] M. Lukoševičius and H. Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.

[7] C. Gallicchio, A. Micheli, and L. Pedrelli. Deep reservoir computing: A critical experimental analysis. *Neurocomputing*, 2016, Accepted.

[8] C. Gallicchio and A. Micheli. Deep reservoir computing: A critical analysis. In *Proc. of the 24th European Symposium on Artificial Neural Networks (ESANN)*, pages 497–502. i6doc.com, 2016.

[9] H. Jaeger, M. Lukoševičius, D. Popovici, and U. Siewert. Optimization and applications of echo state networks with leaky-integrator neurons. *Neural Networks*, 20(3):335–352, 2007.

[10] J. Boedecker, O. Obst, J.T. Lizier, N.M. Mayer, and M. Asada. Information processing in echo state networks at the edge of chaos. *Theory in Biosciences*, 131(3):205–213, 2012.

[11] D. Verstraeten and B. Schrauwen. On the quantification of dynamics in reservoir computing. In *International Conference on Artificial Neural Networks*, pages 985–994. Springer, 2009.

[12] F.M. Bianchi, L. Livi, and C. Alippi. Investigating echo state networks dynamics by means of recurrence analysis. *arXiv preprint arXiv:1601.07381*, pages 1–25, 2016.

[13] H. Jaeger. The ”echo state” approach to analysing and training recurrent neural networks - with an erratum note. Technical report, GMD - German National Research Institute for Computer Science, 2001.