

Article

Echo State Condition at the Critical Point

Norbert Michael Mayer

Department of Electrical Engineering and Advanced Institute of Manufacturing with High-Tech Innovations (AIM-HI), National Chung Cheng University, Chia-Yi 62102, Taiwan; mikemayer@ccu.edu.tw; Tel.: +886-5-272-0411 (ext. 33219)

Academic Editor: Mikhail Prokopenko

Received: 29 October 2016; Accepted: 15 December 2016; Published: 23 December 2016

Abstract: Recurrent networks with transfer functions that fulfil the Lipschitz continuity with $K = 1$ may be echo state networks if certain limitations on the recurrent connectivity are applied. It has been shown that it is sufficient if the largest singular value of the recurrent connectivity is smaller than 1. The main achievement of this paper is a proof under which conditions the network is an echo state network even if the largest singular value is one. It turns out that in this critical case the exact shape of the transfer function plays a decisive role in determining whether the network still fulfills the echo state condition. In addition, several examples with one-neuron networks are outlined to illustrate effects of critical connectivity. Moreover, within the manuscript a mathematical definition for a critical echo state network is suggested.

Keywords: reservoir computing; uniformly state contracting networks; power law

1. Introduction

Classic approaches of recurrent neural networks (RNNs), such as back-propagation through time [1], have been considered difficult to handle. In particular learning in the recurrent layer is slow and problematic due to potential instabilities. About 15 years ago, reservoir computing [2] was suggested as an alternative approach for RNNs. Here, it is not necessary to train connectivity in the recurrent layer. Instead, constant, usually random, connectivity weights are used in the recurrent layer. The supervised learning can be done by training the output layer using linear regression. Two types of reservoir computing are well established in the literature. The first is called liquid state machines (LSMs, [3]), which are usually based on a network of spiking neurons. The second type is called an echo state network (ESN, [4]), which uses real valued neurons that initially use a sigmoid as a transfer function. Although a random recurrent connectivity pattern can be used, heuristically it has been found that typically the performance of the network depends strongly on the statistical features of this random connectivity (cf. for example [5] for ESNs).

Thus, what is a good reservoir with regard to particular stationary input statistics? This has been a fundamental question for research in this field since the invention of reservoir computing. One fundamental idea is that a reservoir can only infer training output from this window of the input history of which traces still can be found inside the reservoir dynamics. However, if the necessary inference from time series in order to learn the training output is far in the past, it may happen that no traces of this input remain inside the reservoir. So, the answer seems to be that a good reservoir is a reservoir from whose states it is possible to reconstruct an input history with a time span that is as long as possible. More precisely, they should be reconstructed in a way that is sufficiently accurate in order to predict the training output. In other words, a good reservoir is a reservoir that has a good memory of the input history.

There have been efforts to quantify the quality of the memory of the reservoir. Most common is the “memory capacity” (MC) according to Jaeger’s definition [4]. However, MC has several drawbacks.

For example, it is not directly compatible to a Shannon information based measure. Still, it illustrates that ESNs are relatively tightly restricted in the way that the upper limit of the MC is equal to the number of hidden layer neurons. So the capabilities of the network increase with the number of neurons.

One more important limiting factor with regard to the reservoir memory is the strength of the recurrent connectivity. According to the echo state condition, the nature of the reservoir requires that the maximum $|\lambda|_{\max}$ of its eigenvalues in modulus is smaller than 1, which is called the echo state property (ESP). This seems always to result in an exponential forgetting of previous states. Thus, forgetting is independent from the input statistics but instead has to be pre-determined and is due to the design of the reservoir dynamics.

In order to proceed, there are several important aspects. First, it is necessary to get rid of the intrinsic time scale of forgetting that is induced by $|\lambda|_{\max} < 1$. More precisely, the remaining activity of inputs to the reservoir that date back earlier than Δt is a fraction smaller than $|\lambda|_{\max}^{\Delta t}$. Networks where the largest eigenvalue is larger than 1 cannot be used as reservoirs anymore, a point which is detailed below. One can try $|\lambda|_{\max} = 1$ and see if this improves the network performance and how this impacts the memory of the reservoir on earlier events. Steps toward this direction have been made by going near the “edge of chaos” [5] or even further where the network may not be an echo state network for all possible input sequences but instead just around some permissible inputs [6]. Presumably, these approaches still all forget exponentially fast.

Strictly, networks with $|\lambda|_{\max} = 1$ are not covered by the initial proof of Jaeger for the echo state condition. One important purpose of this paper is to close this gap and to complete Jaeger’s proof in this sense. The other purpose is to motivate the principles of [7] in as simple as possible examples and thus to increase the resulting insight.

The intentions of the following sections of the paper are to motivate the concept of critical neural networks and explain how they are related to memory compression. These intentions comprise a relatively large part of the paper because it seems important to argue for the principle value of critical ESNs. Section 2 introduces the concept of reservoir computing and also defines important variables for the following sections. An important feature is that Lyapunov coefficients are reviewed in order to suggest a clear definition for critical reservoirs that can be used analytically on candidate reservoirs. Section 3 describes how critical one-neuron reservoirs can be designed and also introduces the concept of extending to large networks. Section 4 explains why the critical ESNs are not covered by Jaeger’s proof. The actual proof for the echo state condition can be found in Section 5. Certain aspects of the proof have been transferred to the appendix.

2. Motivation

The simplest way to train with data in a supervised learning paradigm is to interpolate data (cf. for example [8]). Thus, for a time series of input data $\mathbf{u}_t \in \mathbb{R}^n$ that forms an input sequence $\bar{\mathbf{u}}^\infty$ and a corresponding output data $\mathbf{o}_t \in \mathbb{R}^m$ one can choose a vector of non-linear, linearly independent functions $\mathbf{F}(\mathbf{u}_t) : \mathbb{R}^n \rightarrow \mathbb{R}^k$ and a transfer matrix $\mathbf{w}^{out} : \mathbb{R}^k \rightarrow \mathbb{R}^m$. Then, one can define

$$\begin{aligned}\mathbf{x}_t &= \mathbf{F}(\mathbf{u}_t) \\ \tilde{\mathbf{o}}_t &= \mathbf{w}^{out} \mathbf{x}_t.\end{aligned}$$

\mathbf{w}^{out} can be calculated by linear regression, i.e.,

$$\mathbf{w}^{out} = (A A')^{-1} (A B), \quad (1)$$

where the rectangular matrices $A = [\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_t]$ and $B = [\mathbf{o}_0, \mathbf{o}_1, \dots, \mathbf{o}_t]$ are composed from the data of the training set and A' is the transpose of A . Further, one can use a single transcendental function $\theta(\cdot)$ such that

$$\mathbf{x}_t = \mathbf{F}(\mathbf{u}_t) = \theta(\mathbf{w}^{in} \mathbf{u}_t), \quad (2)$$

where $\mathbf{w}^{in} : \mathbb{R}^n \rightarrow \mathbb{R}^k$ is a matrix in which each line consists of a unique vector and $\theta(\cdot)$ is defined in the Matlab fashion; so the function is applied to each entry of the vector separately. Linear independence of the components of \mathbf{F} can then be guaranteed if the column vectors of \mathbf{w}^{in} are linearly independent. Practically, linear independence can be assumed if the entries of \mathbf{w}^{in} are chosen randomly from a continuous set and $k \geq n$.

The disadvantage of the pure interpolation method with regard to time series is that the input history, that is $\mathbf{u}_{t-1}, \mathbf{u}_{t-2}, \dots, \mathbf{u}_0$, has no impact on training the current output $\hat{\mathbf{o}}_t$. Thus, if a relation between previous inputs and current outputs exists, that relation cannot be learned.

Different from Equation (1), a reservoir in the sense of reservoir computing [2,9,10] can be defined as

$$\mathbf{x}_t = \mathbf{F}(\mathbf{x}_{t-1}, \mathbf{u}_t). \tag{3}$$

The recursive update function adds several new aspects to the interpolation that is outlined in Equation (1):

1. The new function turns the interpolation into a dynamical system:

$$\mathbf{x}_t = \mathbf{F}(\mathbf{x}_{t-1}, \mathbf{u}_t) = \mathcal{F}_{\bar{\mathbf{u}}^\infty}(\mathbf{x}_{t-1}), \tag{4}$$

where the notation $\mathcal{F}_{\bar{\mathbf{u}}^\infty}(\mathbf{x}_{t-1})$ is intended to illustrate the character of the particular discrete time, deterministic dynamical system and the fact that each possible time series $\bar{\mathbf{u}}^\infty$ defines a specific set of dynamics (For obvious reasons one may call \mathbf{F} an input driven system.).

The superset \mathcal{F} over all possible $\bar{\mathbf{u}}^\infty$,

$$\mathcal{F} = \bigcup_{\bar{\mathbf{u}}^\infty} \mathcal{F}_{\bar{\mathbf{u}}^\infty},$$

may be called the reservoir dynamics. Thus, \mathcal{F} covers all possible dynamics of a particular reservoir with regard to any time series of input vectors in \mathbb{R}^n . Note that this way of looking into the dynamics of reservoirs is non-standard. Rather the standard approach is to interpret the reservoir as a non-autonomous dynamical system and then to formalize the system accordingly [6]. For the present work, the turn towards standard dynamical systems has been chosen because here the relevant methodology is well established and the above mentioned formalization appears sufficient for all purposes of this work.

2. It is now possible to account for information from a time series' past in order to calculate the appropriate output.

One important question is if the regression step in the previous section, and thus the interpolation, works at all for the recursive definition in Equation (3). Jaeger showed ([4], p. 43; [11]) that the regression is applicable, i.e., the echo state property (ESP) is fulfilled, if and only if the network is uniformly state contracting. Uniformly state contraction is defined in the following.

Assume an infinite stimulus sequence $\bar{\mathbf{u}}^\infty = \{\mathbf{u}_n\}_{n=0}^\infty$ and two random initial internal states of the system \mathbf{x}_0 and \mathbf{y}_0 . To both initial states \mathbf{x}_0 and \mathbf{y}_0 the sequences $\bar{\mathbf{x}}^\infty = \{\mathbf{x}_n\}_{n=0}^\infty$ and $\bar{\mathbf{y}}^\infty = \{\mathbf{y}_n\}_{n=0}^\infty$ can be respectively assigned.

$$\begin{aligned} \mathbf{x}_t = \mathbf{F}(\mathbf{x}_{t-1}, \mathbf{u}_t) &= \mathcal{F}_{\bar{\mathbf{u}}^\infty}(\mathbf{x}_{t-1}) \\ \mathbf{y}_t = \mathbf{F}(\mathbf{y}_{t-1}, \mathbf{u}_t) &= \mathcal{F}_{\bar{\mathbf{u}}^\infty}(\mathbf{y}_{t-1}) \\ q_{\bar{\mathbf{u}}^\infty, t} &= d(\mathbf{x}_t, \mathbf{y}_t), \end{aligned} \tag{5}$$

where $q_{\bar{\mathbf{u}}^\infty}$ is another series and $d(\cdot, \cdot)$ shall be a distance measure using the square norm.

Then the system \mathcal{F} is uniformly state contracting if it is independent from $\bar{\mathbf{u}}^\infty$ and if for any initial state $(\mathbf{x}_0, \mathbf{y}_0)$ and all real values $\epsilon > 0$ there exists a finite $\tau_{\mathcal{F}}(\epsilon) < \infty$ for which

$$\max_{\mathcal{F}} q_{\bar{\mathbf{u}}^\infty, t} \leq \epsilon \tag{6}$$

for all $t \geq \tau_{\mathcal{F}}$.

Another way to look at the echo state condition is that the network \mathcal{F} behaves in a time invariant manner, in the way that some finite subsequence in an input time series will roughly result always in the same outcome. In other words

$$\mathbf{x}_{\Delta t+t_0} \approx \mathbf{y}_{\Delta t+t_0}$$

independent of t_0 , \mathbf{x}_{t_0} and \mathbf{y}_{t_0} and if Δt is sufficiently large.

Lyapunov analysis is a method to analyze predictability versus instability of a dynamical system (see [12]). More precisely, it measures exponential stability.

In the context of non-autonomous systems, one may define the Lyapunov exponent as

$$\Lambda_{\bar{\mathbf{u}}^\infty} = \lim_{|q_{\bar{\mathbf{u}}^\infty,t=0}| \rightarrow 0} \lim_{t \rightarrow \infty} \frac{1}{t} \log \frac{|q_{\bar{\mathbf{u}}^\infty,t}|}{|q_{\bar{\mathbf{u}}^\infty,0}|} \tag{7}$$

Thus, if

$$q_{\bar{\mathbf{u}}^\infty,t} \propto \exp(bt),$$

then $\Lambda_{\bar{\mathbf{u}}^\infty}$ approximates b and thus measures the exponent of exponential decay. For power law decays, the Lyapunov exponent is always zero (For example, one may try $q_{\bar{\mathbf{u}}^\infty,t} = \frac{1}{t+1}$).

In order to define criticality, we use the following definition.

Definition 1. A reservoir that is uniformly state contracting shall be called critical if for at least one input sequence $\bar{\mathbf{u}}^\infty$ there is at least one Lyapunov exponent $\Lambda_{\bar{\mathbf{u}}^\infty}$ that is zero.

The echo state network (ESN) is an implementation of reservoir dynamics as outlined in Equation (3). Like other reservoir computing approaches, the system is intended to resemble the dynamics of a biologically inspired recurrent neural network. The dynamics can be described for discrete time-steps t , with the following equations:

$$\begin{aligned} \mathbf{x}_{lin,t} &= \mathbf{W}\mathbf{x}_{t-1} + \mathbf{w}^{in}\mathbf{u}_t \\ \mathbf{x}_t &= \theta(\mathbf{x}_{lin,t}) \\ \tilde{\mathbf{o}}_t &= \mathbf{w}^{out}\mathbf{x}_t. \end{aligned} \tag{8}$$

With regard to the transfer function $\theta(\cdot)$, it shall be assumed that it is continuous, differentiable, transcendental and monotonically increasing with the limit $1 \geq \theta'(\cdot) \geq 0$, which is compatible with the requirement that $\theta(\cdot)$ fulfills the Lipschitz continuity with $K = 1$. Jaeger’s approach uses random matrices for \mathbf{W} and \mathbf{w}^{in} , learning is restricted to the output layer \mathbf{w}^{out} . The learning (i.e., training \mathbf{o}_t) can be performed by linear regression (cf. Equation (1)).

The ESN fulfills the echo state condition (i.e., it is uniformly state contracting) if certain restrictions on the connectivity of the recurrent layer apply, for which one can name a necessary condition and a sufficient condition:

- C1 A network has echo states only if

$$1 > |\lambda|_{\max} = \max \text{abs}(\lambda(\mathbf{W})), \tag{9}$$

i.e., the absolute value of the biggest eigenvalue of \mathbf{W} is below 1. The condition means that a network is **not** and ESN if $1 < \max \text{abs}(\lambda(\mathbf{W}))$.

- C2 Jaeger named here initially

$$1 > s_{\max} = \max s(\mathbf{W}), \tag{10}$$

where s is the vector of singular values of the matrix \mathbf{W} . However, a closer sufficient condition has been found in [13]. Thus, it is already sufficient to find a full rank matrix D for which

$$1 > \max s(D\mathbf{W}D^{-1}). \tag{11}$$

The authors of [14] found another formulation of the same constraint: *The network with internal weight matrix \mathbf{W} satisfies the echo state property for any input if \mathbf{W} is diagonally Schur stable, i.e., there exists a diagonal $P > 0$ such that $\mathbf{W}^T P \mathbf{W} - P$ is negative definite.*

Apart from the requirement that a reservoir has to be uniformly state contracting, the learning process itself is not of interest in the scope of this paper.

3. Critical Reservoirs with Regard to the Input Statistics \bar{u}^∞

Various ideas on what types of reservoirs work better than others have been brought up. One can try to keep the memories of the input history in the network as long as possible. The principle idea is to tune the network’s recurrent connectivity to a level where the convergence for a subset of $\mathcal{F}_{crit} \in \mathcal{F}$ with regard to Equation (6) is

$$q_{\bar{u}_{crit}^\infty, t} \propto t^a \tag{12}$$

rather than

$$q_{\bar{u}^\infty, t} \propto b^t, \tag{13}$$

where $a < 0$ and $0 < b < 1$ are system specific values, i.e., they depend on $\mathcal{F}_{\bar{u}^\infty}$.

A network according to Equation (12) is still an ESN since it fulfills the ESP. Still, forgetting of initial states is not bound to a certain time scale. Remnants of information can —under certain circumstances— remain for virtually infinite long times within the network given that not too much unpredictable new input enters the network. Lyapunov analysis of a time series according to Equation (12) would result in zero, and Lyapunov analysis of Equation (13) yields a nonzero result.

In ESNs forgetting according to the power law of Equation (12) for an input time series $q_{\bar{u}_{crit}^\infty, t}$ is achievable if the following constraints are fulfilled:

- The recurrent connectivity \mathbf{W} , the input connectivity \mathbf{w}^{in} of the ESN and the transfer function $\theta(\cdot)$ have to be arranged in a way that if the ESN is fed with $q_{\bar{u}_{crit}^\infty, t}$ one approximates

$$\lim_{t \rightarrow \infty} |\dot{\theta}(\mathbf{x}_{lin, t})| = 1.$$

Thus, the aim of the training is

$$|\dot{\theta}(\mathbf{x}_{lin, t})| = 1. \tag{14}$$

Since the ESN has to fulfil the Lipschitz continuity with $K = 1$, the points where $\dot{\theta} = 1$ have to be inflection points of the transfer function. In the following these inflection points shall be called epi-critical points (ECPs).

- The recurrent connectivity of the network is to be designed in a way that the largest absolute eigenvalue and the largest singular value of \mathbf{W} both are equal to one. This can be done by using normal matrices for \mathbf{W} (see Section 3.5).

3.1. Transfer Functions

The standard transfer function that is widely used in ESNs and other types of neural networks is the sigmoid function, i.e.,

$$\theta(x) = \tanh(x). \tag{15}$$

The sigmoid transfer function has one ECP $\Pi_0 = 0$.

As a second possible transfer function, one may consider

$$\theta(x) = 0.5x - 0.25 \sin(2x). \quad (16)$$

Here, one has a infinite set of ECPs at $\Pi_i = (n + 0.5) \times \pi$. It is important to have more than one ECP in the transfer functions because for a network with discrete values it appears necessary that each neuron has at least 2 possible states in order to have any inference from the past to the future. In the case of tanh, the only solution of Equation (14) is

$$\mathbf{x}_{lin,t} = 0.$$

That type of trained network cannot infer information from the past to the present for the expected input, which significantly restricts its capabilities. One can see that from

$$\mathbf{x}_t = \tanh(\mathbf{x}_{lin,t}) = 0.$$

The next iteration yields

$$\mathbf{x}_{t+1} = \tanh(\mathbf{w}^{in} \mathbf{u}_t),$$

which again, after training would require

$$0 = \mathbf{w}^{in} \mathbf{u}_t.$$

Thus, the network can only anticipate the same input at all time steps.

In the case of Equation (16), the maximal derivative $\theta'(x)$ is 1 at $x = \pi(n + 1/2)$, where n is an integer number (confer Figure 1). Here the main advantage is that there exists an infinite set of epi-critical points. However, all these points are positioned along the linear function $y = x/2$. This setting still significantly restricts the training of \mathcal{F}_{crit} . Here one can consider the polynomial with the lowest possible rank (cf. the green line in Figure 1, left side) that interpolates between the epi-critical points (in the following called an epi-critical transfer function). In the case of Equation (16) the epi-critical transfer function is the linear function

$$\mathbf{x}_{t+1} = 0.5\mathbf{W}\mathbf{x}_t + 0.5\mathbf{w}^{in} \mathbf{u}_t.$$

Thus, the effective dynamics of the trained reservoir on the expected input time series is -if this is possible- the dynamics of a linear network. This results in a very restricted set of trainable time series.

As an alternative, one could consider also a transcendental function for the interpolation between the points, such as depicted in Figure 2. The true transfer function (blue line in Figure 2) can be constructed in the following way. Around a set of defined epi-critical points Π_i , define θ as either

$$\theta(x) = \tanh(x - \Pi_i) + \tanh(\Pi_i)$$

or

$$\theta(x) = \tanh(x).$$

This is one conceptional suggestion for further investigations. The result is a transfer function with the epi-critical points Π_i and 0. The epi-critical transfer in this case is a tanh function.

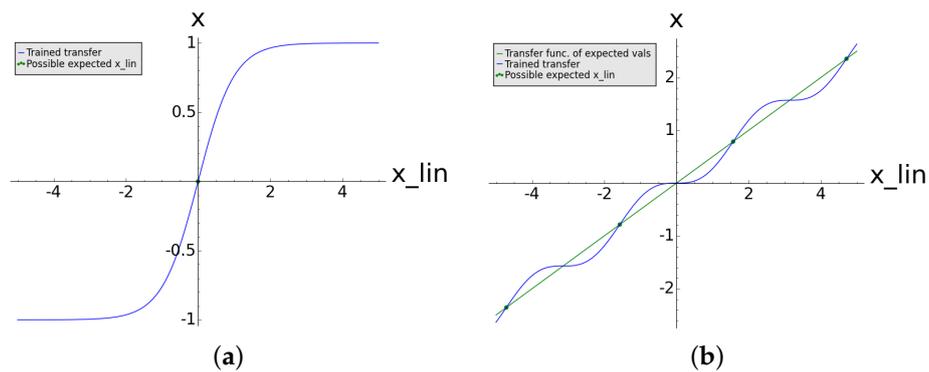


Figure 1. Variant possible transfer functions with interesting features regarding criticality. On the left side (a) tanh has only one epi-critical point. Equation (16) is graphed on the right side (b). Here, there is an infinite set of epi-critical points $n\pi + \pi/2$ that are all positioned along the line $y = x/2$. In both graphs green dots indicate epi-critical points, green curves are smooth interpolations between those points, and the blue line indicates the particular transfer function itself.

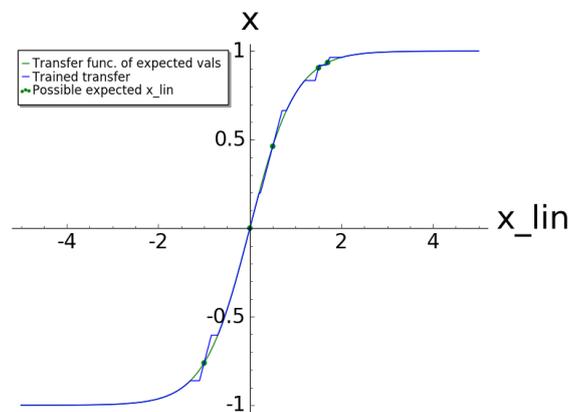


Figure 2. Tailored transfer function where the epi-critical points can be organized in an adaptive way and follow a transcendental function (which has certain advantages to the examples depicted in Figure 1, in blue). Green dots indicate epi-critical points, the green curve is a smooth interpolation between those points, and the blue line indicates the particular transfer function itself. Note that the green curve is a tanh and thus a transcendental function.

3.2. Examples Using a Single Neuron as a Reservoir

In this and the following sections, practical examples are brought up where a single neuron represents a reservoir. Single neuron reservoirs have been studied in other researches [15,16]. Here the intention is to illustrate the principle benefits and other features of critical ESNs.

First one can consider a neuron with tanh as a transfer function along with a single input unit

$$x_{t+1} = \tanh(bx_t + u_t), \tag{17}$$

where in order to achieve a critical network $|b|$ has to be equal to 1. i.e., the network exactly fulfills the boundary condition. From previous consideration one knows that \mathcal{F}_{crit} has the dynamics that results from $u_t = 0$ as an input. In this case the only fixed-point of the dynamics is $x_t = 0$, which is also the epi-critical point if $|b| = 1$.

Power law forgetting: Starting from the two initial values $x_0 = 0$ and say $y_0 = 0.01$, one can see that the two networks converge in a power law manner to zero. On the other hand, for a linear network

$$x_{t+1} = bx_t + u_t$$

with the same initial conditions the dynamics of the two networks never converge (independently of u_t). Instead, the difference between $x_0 = 0$ and say $y_0 = 0.01$ stays the same forever. Thus, the network behavior in the the case of $|b| = 1$ depends on the nature of the transfer function. For all other values of b , both transfer functions result qualitatively in the same behavior in dependence on b : either they diverge or they converge exponentially. Since $|b| = 1$ is also the border between convergence and divergence and thus the border between uniformly state contracting networks and not uniformly state contracting networks, the case of $|b| = 1$ is a critical point of the dynamical system, in a similar manner as a critical point at the transition from ordered dynamics to instability. In the following it is intended to extend rules for different transfer functions, where different transfer functions result in the critical point in uniformly state contracting networks and where this is not the case.

As a final preliminary remark, it has to be emphasized that a network being uniformly state contracting means that the states are contracting for any kind of input u_t . It does not mean that for any kind of input the contraction follows a power law. In fact, for all input settings $u_t \neq 0$ the contraction is exponential for the neuron of Equation (17), even in the critical case ($|b| = 1$).

3.3. Single Neuron Network Example with Alternating Input and Power Law Forgetting

As outlined above, for practical purposes the sigmoid function, i.e., $\theta = \tanh(\cdot)$, is not useful for critical networks because the only critical state occurs when total activity of such a network is null. In that case it is not possible to transfer information about the input history. The reason is illustrated in the following example, where instead of a sigmoid function other types of transfer functions are used.

So, the one-neuron network

$$x_{t+1} = \theta(-bx_t + (2 - b)u_t) \tag{18}$$

with a constant b , the transfer function of Equation (16) and the expected alternating input $u_t = (-1)^t \pi/4$ has an attractor state when also x_t is alternating with $x_t = (-1)^t \pi/4$ independently from b . Thus, $u_t = (-1)^t \pi/4$ shall be interpreted as the expected input that directs the activity of the network exactly to where $\theta' = 1$ and thus induces a critical dynamic $\mathcal{F}_{\bar{u}^\infty} = \mathcal{F}_{crit}$.

It is now interesting to investigate the convergence behavior for different values of b considering differing starting values for internal states x_t and y_t . Figure 3 depicts the resulting different Lyapunov exponents for one-neuron networks with different values of b . One can see that—not surprisingly—the Lyapunov exponent for $b = 1$ is zero. This is the critical point that marks the transition from order to instability in the system and at the same time the transition from ESP to non-ESP.

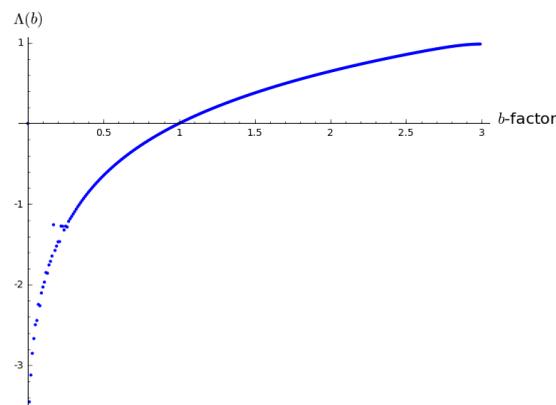


Figure 3. Depicted is the Lyapunov exponent for the example system of Equation (18) for different values of b . At $b = 1$ the Lyapunov exponent crosses zero. At this point all types of linear analysis fail. The point marks the border between networks that fulfil the echo state property (ESP) and those that do not. So, this point is the called critical point. Further analysis shows that the point itself belongs to the area with ESP. All results from Figures 4 and 5 are drawn from the point where the Lyapunov exponent is zero, that is $b = 1$.

In the following the same network at $b = 1$ is investigated. Some unexpected input, i.e., $u_t \neq (-1)^t \pi/4$, lets the network jump out of the attractor. If the input afterwards continues as expected, the network slowly returns to the attractor state in a power law fashion. Thus,

$$d(x_t, y_t) \propto t^{-a},$$

where y_t represents the undisturbed time series and a is a constant value. Note that x_t contains all information of the network history and that the network was simulated using IEEE 754-2008 double precision variables, which have a memory size of 64 bits on Intel architecture computers. Although floating point variables are organized in a complicated way of three parts, the sign, exponent and mantissa, it is clear that the total reservoir capacity cannot exceed those 64 bits, which means in the limit a reservoir of one-neuron cannot remember more than 64 binary i.i.d. random numbers.

Thus, about 64 iterations after an unexpected input the difference between x_t and y_t should be annihilated. Thus, if both networks x_t and y_t receive the same unexpected input that is of the same magnitude, the difference between the x_t and y_t should reach virtually 0 within 64 iterations. The consideration can be tested by setting the input

$$u_t = \pi/4 \times \text{rand},$$

where rand is an i.i.d. random list of +1s and -1s, that produces a representative of $\mathcal{F}_{\bar{u}^\infty} \neq \mathcal{F}_{\text{crit}}$. Figures 4 and 5 depict results from simulations, where after one iteration when two networks receive different inputs, both networks receive again the same input. Depicted is again the development of the difference between both networks $d(x_t, y_t)$ versus the number of iterations. The graphs appear in a double logarithmic fashion, so power law decays appear as straight lines. One can see that the networks that receive alternating, i.e., the expected input, pertains the difference for very long time spans (that exceed 64 iterations). On the other hand, if the network input for both networks is identical but i.i.d random and of the same order of magnitude as the expected input, the difference vanishes within 64 iterations.

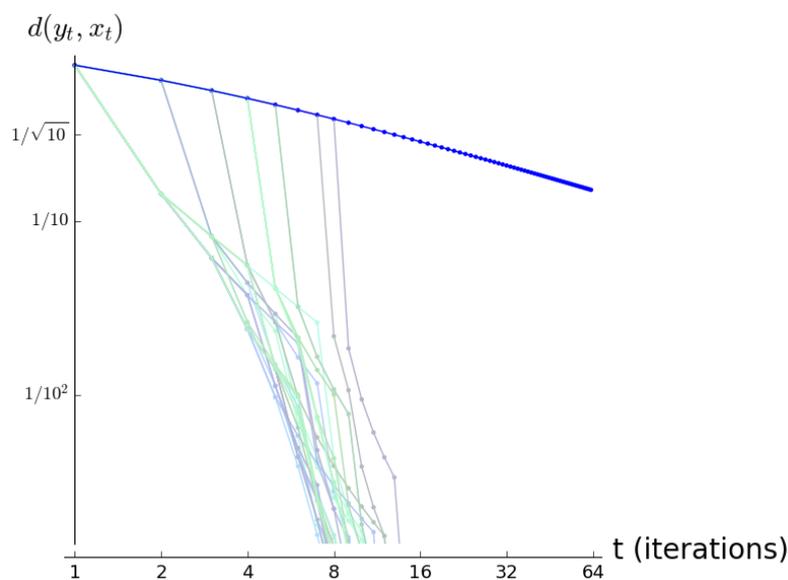


Figure 4. Results from two dynamics defined in Equation (18), with $b = 1$ and different types on input u_t . One copy of the two initially identical networks has received a variant input at iteration one. Depicted here is the decay of the difference of the state in the recurrent layer if both networks receive the same and expected (alternating) input (in dark blue). The pale curves are data from Figure 5 embedded for comparison. One can see that the difference function (blue) follows a power law and pertains for longer than 64 iterations.

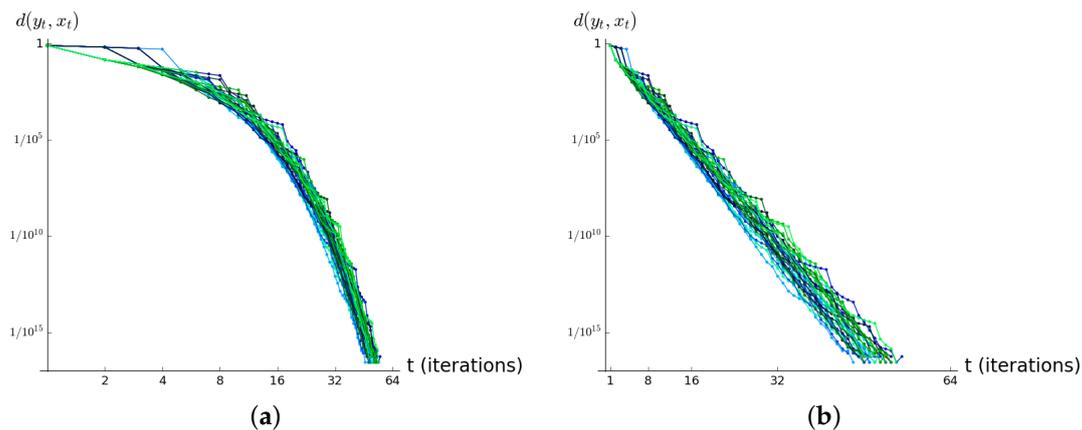


Figure 5. Complementary data for Figure 4: (a) Decay if both networks receive the same but irregular (i.i.d random) input (different colors are used for different trials). The difference vanishes much faster than in the left case. Finally the difference becomes and stays null, which is out of a logarithmic scale. Thus, later iterations are not depicted anymore; (b) Depicted here is the same data as at the left, however in a double-logarithmic plot. The data forms a straight line, which indicates an exponential decay.

The network distinguishes regular input from irregular input. Memories of irregular events pertain for a long time in the network provided that the following input is regular again. How a reservoir can be trained to anticipate certain input statistics has been discussed [7]. Additional solutions to this problem are subject to further investigations.

3.4. Relation to “Near Edge of Chaos” Approaches

It is a common experience that—in spite of given theoretical limits for the ESPs—the recurrent connectivity can be significantly higher than $s_{\max} = 1$ for many practical input statistics. Those over-tuned ESNs in many cases show a much better performance than those that actually obey Jaeger’s initial limit ESP. So, recently researchers came up with theoretical insights with regard to ESPs that are subject to a network and a particular input statistic [6]. In the scope of this work, instead of defining the ESP for a network the ESP is always defined as related to a network and an input statistic. Also, similar efforts have been undertaken in the field of so-called liquid state machines [17,18].

One may assume that those approaches show similar properties as the one that has been presented here. However, for a good reason those approaches all are called ‘near edge of chaos’ approaches. In order to illustrate the problems that arise from those approaches, one may consider what happens if those overtuned ESNs are set exactly to the critical point. Here, just for the general understanding one may consider again a one-neuron network and a tanh as a transfer function, so

$$x_{t+1} = \tanh(-bx_t + u_t). \quad (19)$$

Note that the ESP limit outlined above requires that the recurrent connectivity should be $b < 1$. An input time series than one can use is from the previous section

$$u_t = (-1)^t \pi/4 \quad (20)$$

Slightly tedious but basically simple calculus results in a critical value of $b \approx 2.344$ for the input time series, where $x_t \approx (-1)^t \times 0.757$. In this situation one can test for the convergence of two slightly different initial conditions and obtain a power law decay of the difference. However, setting up the amplitude of the input just a tiny bit higher is going to result in two diverging time series x_t and y_t . If the conditions of the ESN are chosen to be exactly at the critical point, it is possible that a untrained input sequence very close to the trained input sequence turns the ESN into a state where the ESP is not fulfilled anymore (for a related and more detailed discussion with a numerical analysis, confer [19]).

For this reason all the networks have to be chosen at a significant margin away from the edge of instability. That is very different from the approach in the previous section where although the expected input sequence for the network is exactly at the critical point, all other input sequences result in a stable ESN in most cases with exponential forgetting.

3.5. How the One-Neuron Example Can Be Extended to Multi-Neuron Networks: Normal Matrices

A normal matrix \mathbf{W} commutes with its own transpose, i.e.,

$$\mathbf{W}\mathbf{W}^T = \mathbf{W}^T\mathbf{W}.$$

For a normal matrix \mathbf{W} , it can easily be shown that

$$\max s(\mathbf{W}) = \max \text{abs}(\lambda(\mathbf{W})).$$

These matrices apply to the spectral theorem; the largest absolute eigenvalue is the same as the largest singular value, which makes a clear theoretical separation between networks that are uniformly state contracting and those that are not compatible to the echo state condition. Still, for normal matrices all previously known ES conditions do not determine to which of those two groups the critical point itself belongs.

Summarizing, all previous works result in theorems for an open set of conditions that are defined by the strict inequalities Equations (9) and (11). In the closest case, the case of normal matrices, when considering the singular condition

$$1 = \max s(\mathbf{W}) = \max \text{abs}(\lambda(\mathbf{W})) \quad (21)$$

there is no statement of the above mentioned theorems if the network is uniformly state contracting.

Some simple, preliminary numerical tests reveal that in the case of networks that satisfy Equation (21) the further development of the network strictly depends on the exact shape of transfer function.

4. Echo State Condition Limit with Weak Contraction

In the previous section, it has been shown how power law forgetting may occur in an ESN type neural network. These networks are all tuned to the point where $|\lambda|_{\max} = S = 1$. For this tuning it is still undetermined if the ESP is fulfilled or not even if normal matrices or one-neuron RNNs are used. The current section is dedicated to determining under which conditions Jaeger's ESP can be extended to this boundary condition.

Jaeger's sufficient echo state condition (see [4], App. C, p. 41) has strictly been proven only for non-critical systems (largest singular value $S < 1$) and with $\tanh(\cdot)$ as a transfer function. The original proof is based on the fact that \tanh in combination with $S < 1$ is a contraction. In that case Jaeger shows an exponential convergence.

The core of all considerations of a sufficient condition is to give a close upper estimate of the distance between the next iterations of two different states y_t and x_t . The estimate is of the form

$$\max_{\mathbf{u}^\infty} d(y_{t+1}, x_{t+1}) = \max_{\mathcal{F}} d(\mathcal{F}(y_t), \mathcal{F}(x_t)) \leq \phi_1 \cdot d(y_t, x_t),$$

where the parameter ϕ_1 basically is quantified by the nature of the transfer function and the strength of the connectivity matrix. The estimate has to be good enough that the iterative application of ϕ_1 should result in a convergence to 0:

$$\lim_{t \rightarrow \infty} [\phi_1 \cdot]^t d(y_0, x_0) = 0, \quad (22)$$

This is equivalent to investigating a series q_t with

$$d(y_t, x_t) < q_t = [\phi_1 \cdot]^t d(y_0, x_0)$$

and

$$\lim_{t \rightarrow \infty} q_t = 0, \tag{23}$$

which can prove that the requirement of uniformly state contraction (cf. Equation (6)) is fulfilled. For example, consider the case of a reservoir with one-neuron as described in Equation (17). Here the challenge is to find an estimator for ϕ_1 such that

$$\max_{u_t \in \mathbb{R}} d(\theta(by_t + u_t), \theta(bx_t + u_t)) \leq \phi_1 \cdot d(y_t, x_t),$$

where the chosen ϕ_1 still holds the limit in Equation (22). For $|b| < 1$, convergence can be proven easily:

$$\begin{aligned} \max_{u_t \in \mathbb{R}} d(\theta(by_t + u_t), \theta(bx_t + u_t)) &\leq \\ \max_{u_t \in \mathbb{R}} \|by_t + u_t - bx_t - u_t\|_2 &= \\ |b| \cdot d(y_t, x_t). \end{aligned}$$

Thus for $|b| < 1$, one can easily define $\phi_1 = |b|$. So q_t can be defined as

$$q_t = |b|^t \cdot d(y_0, x_0).$$

So Equation (23) is fulfilled. The convergence is exponential. The arguments so far are analogous to the core of Jaeger’s proof for the sufficient echo state condition C2 that is restricted to one dimension.

For $|b| = 1$, this argument does not work anymore. Obviously, Jaeger’s proof is not valid under these circumstances. However, the initial theorem can be extended. As a pre-requisite, one can replace the constant ϕ_1 with a function that depends on q_t as an argument.

So one can try

$$\phi_1(q_t) = 1 - \eta q_t^\kappa,$$

where $\eta > 0$ and $\kappa > 1$ have to be defined appropriately. This works for small values of q_t . However, it is necessary to name a limit for large $q_t > \gamma$. Define

$$\phi_1(z) := \begin{cases} 1 - \eta z^\kappa & \text{if } z < \gamma \\ 1 - \eta \gamma^\kappa & \text{if } z \geq \gamma. \end{cases} \tag{24}$$

Three things have to be done to check this cover function:

- First of all, one needs to find out if indeed the cover function ϕ_1 fulfills

$$\max_{u_t \in \mathbb{R}} d(\theta(y_t + u_t), \theta(x_t + u_t)) \leq d(y_t, x_t) \phi_1(d(y_t, x_t)).$$

In order to keep the proof compatible with the proof for multiple neurons for this work, one has to chose a slightly different application for ϕ_1 ,

$$\max_{u_t \in \mathbb{R}} d^2(\theta(y_t + u_t), \theta(x_t + u_t)) \leq d^2(y_t, x_t) \phi_1(d^2(y_t, x_t)),$$

which serves the same purpose and is much more convenient for multiple neurons.

In Appendix C one can find a recipe for this check.

- Second, one has to look for the convergence of

$$q_{t+1} = q_t \phi_1(q_t) = q_t (1 - \eta q_t^\kappa),$$

when $q_t \leq \gamma$. The analysis is done in Appendix A.

- Third, one needs to check

$$q_{t+1} = q_t \phi_1(q_t) = q_t (1 - \eta \gamma^\kappa),$$

as long as $q_t > \gamma$. Since the factor $(1 - \eta \gamma^\kappa)$ is positive, smaller than one and constant, the convergence process is exponential, obviously.

Note that the next section's usage of the cover function differs slightly even through it has the same form as Equation (24).

5. Sufficient Condition for a Critical ESN

The content of this section is a replacement of the condition C2 where the validity of the ESP is inferred for $S \leq 1$.

Theorem 1. *If hyperbolic tangent or the function of Equation (16) are used as transfer functions, the echo state condition (see Equation (6)) is fulfilled even if $S = 1$.*

Summary of the Proof. As an important precondition, the proof requires that both transfer functions fulfill

$$d(\theta(y_t), \theta(x_t)) \leq d(y_t, x_t) \phi_k(d^2(y_t, x_t)), \tag{25}$$

where $\phi_k(z)$ is defined for a network with k hidden neurons as

$$\phi_k(z) := \begin{cases} 1 - \eta z^\kappa & \text{if } z < \gamma \\ 1 - \eta \gamma^\kappa & \text{if } z \geq \gamma \end{cases}. \tag{26}$$

Here, $1 > \gamma > 0, 1 > \eta > 0, \kappa \geq 1$ are constant parameters that are determined by the transfer function and the metric norm $d(\cdot, \cdot) = \|\cdot\|_2$ □

In Appendix C it is shown that indeed both transfer functions fulfil that requirement. It then remains to prove that in the slowest case we have a convergence in a process with 2 stages. In the first stage, if $d^2(y_t, x_t) > \gamma$ there is a convergence that is faster or equal to an exponential decay. The second stage is a convergence process that is faster or equal to a power law decay.

Proof. Note with regard to the test function ϕ_k :

$$\begin{aligned} & \phi_k \leq 1, \\ & \forall z, \forall Z : 0 \leq z \leq Z \quad \leftrightarrow \quad \phi_k(z) \geq \phi_k(Z), \\ & \text{and } \forall z, \forall Z : 0 \leq z \leq Z \quad \leftrightarrow \quad Z \times \phi_k(Z) \geq z \times \phi_k(z) \end{aligned}$$

In analogy to Jaeger, one can check now the contraction between the time step t and $t + 1$:

$$\begin{aligned} d^2(\mathbf{y}_{t+1}, \mathbf{x}_{t+1}) &= d^2(\theta(\mathbf{y}_{lin,t+1}), \theta(\mathbf{x}_{lin,t+1})) \\ &\leq d^2(\mathbf{y}_{lin,t+1}, \mathbf{x}_{lin,t+1}) \times \phi_k(d^2(\mathbf{y}_{lin,t+1}, \mathbf{x}_{lin,t+1})) \end{aligned} \tag{27}$$

One can rewrite

$$d^2(\mathbf{y}_{lin,t+1}, \mathbf{x}_{lin,t+1}) = \|\mathbf{W}\mathbf{y}_t + I - \mathbf{W}\mathbf{x}_t - I\|_2^2 = \|\mathbf{W}(\mathbf{y}_t - \mathbf{x}_t)\|_2^2, \tag{28}$$

where $I = \mathbf{w}^{in} \mathbf{u}_t$. Next one can consider that one can decompose the recurrent matrix by using singular value decomposition (SVD) and obtain $\mathbf{W} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T$. Note that both \mathbf{U} and \mathbf{V} are orthogonal matrices and that \mathbf{S} is diagonal with positive values s_i on the main diagonal. We consider

$$\mathbf{a} = \mathbf{V}^T(\mathbf{y}_t - \mathbf{x}_t).$$

Because \mathbf{V} is an orthogonal matrix, the left side of the equation above is a rotation of the right side and the length $\|\mathbf{a}\|$ is the same as $\|\mathbf{y}_t - \mathbf{x}_t\|$. One can write

$$d^2(\mathbf{y}_t, \mathbf{x}_t) = \sum_i a_i^2,$$

where the a_i are entries of the vector \mathbf{a} . Since

$$\mathbf{y}_{lin,t+1} - \mathbf{x}_{lin,t+1} = \mathbf{U}\mathbf{S}\mathbf{a}$$

and \mathbf{U} is again a rotation matrix, one can write

$$d^2(\mathbf{y}_{lin,t+1}, \mathbf{x}_{lin,t+1}) = \sum_i s_i^2 a_i^2,$$

where s_i is the i -th component of the diagonal matrix \mathbf{S} , i.e., the i -th singular value.

In the following we define $s_{\max} = \max_i s_i$ and calculate

$$\begin{aligned} & d^2(\mathbf{y}_{lin,t+1}, \mathbf{x}_{lin,t+1}) \times \phi_k(d^2(\mathbf{y}_{lin,t+1}, \mathbf{x}_{lin,t+1})) \\ &= (\sum_i s_i^2 a_i^2) \times \phi_k(\sum_i s_i^2 a_i^2) \\ &\leq (s_{\max}^2 \sum_i a_i^2) \times \phi_k(s_{\max}^2 \sum_i a_i^2) \\ &\leq (s_{\max}^2 d^2(\mathbf{y}_t, \mathbf{x}_t)) \times \phi_k(s_{\max}^2 d^2(\mathbf{y}_t, \mathbf{x}_t)) \end{aligned} \tag{29}$$

Merging Equations (27) and (29) results in the inequality

$$d(\mathbf{y}_{t+1}, \mathbf{x}_{t+1}) \leq s_{\max} d(\mathbf{y}_t, \mathbf{x}_t) \times (\phi_k(s_{\max}^2 d^2(\mathbf{y}_t, \mathbf{x}_t)))^{0.5}.$$

First, assuming $s_{\max} < 1$ and since we know $\phi_k \leq 1$, we get an exponential decay

$$d(\mathbf{y}_n, \mathbf{x}_n) \leq s_{\max}^t d(\mathbf{y}_0, \mathbf{x}_0).$$

This case is handled by Jaeger’s initial proof. With regard to an upper limit of the contraction speed (cf. Equation (6)), one can find

$$\tau(\epsilon) = \frac{\log \epsilon - \log(d(\mathbf{y}_0, \mathbf{x}_0))}{\log s_{\max}}.$$

If the largest singular value $s_{\max} > 1$, then for some type of connectivities (i.e., normal matrices) the largest absolute eigenvalue is also larger than 1 due to the spectral theorem. In this case, the echo state condition is not always fulfilled, which has been shown also by Jaeger.

What remains is to check the critical case $s_{\max} = 1$. Here again one can discuss two different situations (rather two separate phases of the convergence process) separately:

If $d^2(\mathbf{y}_{lin,t}, \mathbf{x}_{lin,t}) > \gamma$, we can write the update inequality of Equation (30) as:

$$d^2(\mathbf{y}_{t+1}, \mathbf{x}_{t+1}) \leq (1 - \eta \gamma^\kappa) d^2(\mathbf{y}_t, \mathbf{x}_t).$$

Thus, for all $\epsilon^2 \geq \gamma$, the slowest decay process can be covered by

$$\tau(\epsilon) = \frac{2 \log \epsilon - 2 \log(d(\mathbf{y}_0, \mathbf{x}_0))}{\log(1 - \eta \gamma^\kappa)}.$$

If $\epsilon^2 < \gamma$, then Equation (30) becomes:

$$d^2(\mathbf{y}_{t+1}, \mathbf{x}_{t+1}) \leq d^2(\mathbf{y}_t, \mathbf{x}_t)(1 - \eta d(\mathbf{y}_t, \mathbf{x}_t)^{2\kappa}).$$

One can replace

$$q_t = d^2(\mathbf{y}_t, \mathbf{x}_t) \quad (30)$$

and again consider the sequence

$$q_{t+1} = q_t(1 - \eta q_t^\kappa),$$

which is discussed in Appendix A. The result there is that the sequence converges faster than

$$q_*(t) = \left[\frac{\eta}{\kappa}t + q_0^{-\kappa}\right]^{-1/\kappa}. \quad (31)$$

Note that, although the Lyapunov exponent (cf. Equation (7)) of $q_*(t)$ is zero, the sequence $q_*(t)$ converges in a power law fashion. Thus,

$$\lim_{t \rightarrow \infty} q(t) = d^2(\mathbf{y}_t, \mathbf{x}_t) = 0 \quad (32)$$

and thus ESP has been proven. \square

Moreover, one can calculate the upper time limit $\tau(\epsilon)$:

$$\tau(\epsilon) = \frac{\kappa}{\eta}(\epsilon^{-2\kappa} - a_0^{-\kappa}).$$

6. Summary, Discussion and Outlook

The background of this paper is to investigate the limit of recurrent connectivity in ESNs. The preliminary hypothesis towards the main work can be summarized in Figure 6. Initially it is hard to quantify the transition point between uniformly state contracting and non-state contracting ESNs exactly. However, for normal matrices and one-neuron networks the gap between the sufficient condition and the necessary condition collapses in a way that there are two neighboring open sets. The first open set is known to have the ESP, and the other open set evidently does not have the ESP. What remains is the boundary set. The boundary set is interesting to analyze because it can easily be shown that here power law forgetting can occur.

The proof of Section 4 shows a network is an ESN even if the largest eigenvalue of the recurrent connectivity matrix is equal to 1 and if the transfer function is either Equation (15) or (16). The proof is also extensible to other transfer functions. On the other hand it is obvious that some transfer functions result in networks that are not ESNs. For example a linear transfer function ($\theta(x) = x$) is *not* state contracting.

Even if the network is state contracting, it is not necessarily exponentially uniformly state contracting. Its rate of convergence might follow a power law in the slowest case. Several examples for power law forgetting have been shown in the present work. More examples of preliminary learning have been outlined in [7]. One important target of the present research is to allow for a kind of memory compression in the reservoir by letting only the unpredicted input enter the reservoir.

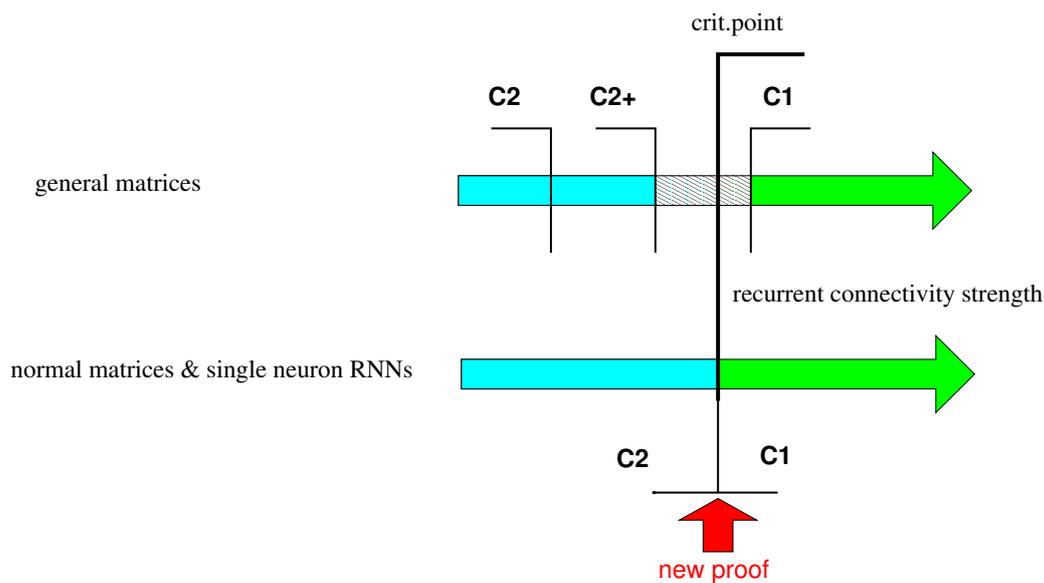


Figure 6. The graph illustrates the relation between the different conditions with regard to echo state property (ESP). C1 and C2 are the necessary and sufficient condition according to Jäger [4]. The large arrows represent the connectivity strength of the recurrent synaptic weight matrix \mathbf{W} . For a small connectivity strength the ESP is fulfilled (cyan areas). For a strong connectivity, the ESP is not fulfilled anymore (green). C2+ represents symbolically the improvement of the sufficient condition according to [13,14]. For general matrices there can be a non-zero gap between C2+ and C1 which is drawn diagonally shaded. The transition from ESP to non-ESP happens somewhere within this gap. All three conditions C1, C2 and C2+ describe an open set. In the case of normal matrices and for one-neuron networks the gap is closed except for the separation line itself. The proof of Section 4 shows that there it depends on the transfer function if the network is an ESN or not.

One ultimate target of the present work is to find a way to organize reservoirs as recurrent filters with a memory compression feature. In order to bring concepts of data compression into the field of reservoir computing and in order to project as much as possible of the input history to the limit size reservoir, principles of memory compression have to be transferred into reservoir computing. However, the reservoir computing techniques that are analogous to classic memory compression have not been identified so far.

Another topic that needs further investigation is entropy in time series. Power law forgetting is only possible if the time series that relates to the criticality is either of a finite entropy, i.e., from a certain point in time all following entries of the time series can be predicted from the previous entries, or if the network simply ignores certain aspects of the incoming time series.

There also potential analogies in biology. Several measurements of memory decay in humans exist that reveal that there the forgetting follows a power law at least for a large fraction of the investigated examples [20,21].

Acknowledgments: The Nation Science Council and the Ministry of Science and Technology of Taiwan provided the budget for our laboratory. Also thanks go to AIM-HI for various ways of support.

Conflicts of Interest: The author declares no conflict of interest.

Notations in Formulas

$s(M)$		vector of singular values of matrix M
\mathbf{S}		diagonal matrix of singular values
s_{\max}		largest singular value of a matrix
$\lambda(M)$		vector of eigenvalues
$ \lambda _{\max}$		the maximum of the eigenvalues in modulus
$\theta(\cdot)$	$\mathbb{R} \rightarrow \mathbb{R}$	transcendental transfer function
$\theta', \dot{\theta}$		its derivative
\mathbf{w}^{in}	$\mathbb{R}^{n \times k}$	input matrix
\mathbf{W}	$\mathbb{R}^{k \times k}$	recurrent transfer matrix
\mathbf{w}^{out}	$\mathbb{R}^{k \times m}$	output matrix
\mathbf{u}_t	\mathbb{R}^n	input time series
$\bar{\mathbf{u}}^\infty$		complete infinite input time series
\mathbf{x}_t	\mathbb{R}^k	hidden layer states
\mathbf{y}_t	\mathbb{R}^k	alternative hidden layer states to check convergence
\mathbf{o}_t		training set
$\hat{\mathbf{o}}_t$		trained network output
$d(\cdot, \cdot)$	$\ \cdot\ _2$	distance measure that is used to check convergence
$\mathbf{F}(\cdot)$		vector of a set of linearly independent, non-linear functions
$\mathcal{F}_{\bar{\mathbf{u}}^\infty}$	$\in \mathcal{F}$	dynamics of the network with regard to the input
\mathcal{F}_{crit}		all dynamics that show power law forgetting
$q_{\bar{\mathbf{u}}^\infty, t}$		time series related to forgetting in ESNs
q_t		time series variable
$\phi_1(\cdot)$		cover function to estimate contraction in a 1 neuron net
$\phi_k(\cdot)$		dto. for a k neuron network
η, κ, γ		const parameters of both ϕ_1 and ϕ_k
$\Lambda_{\bar{\mathbf{u}}^\infty}$		Lyapunov exponent of a neural network wrt. an input time series $\bar{\mathbf{u}}^\infty$

Appendix A. Analyze $q_{t+1} = q_t(1 - \eta q_t^\kappa)$

We can consider the sequence q_t :

$$q_{t+1} = q_t(1 - \eta q_t^\kappa). \quad (\text{A1})$$

Convergence can be analyzed in the following way:

$$\Delta q_t = q_{t+1} - q_t = -\eta q_t^{\kappa+1}. \quad (\text{A2})$$

Thus, the series q_{t+1} can be written as

$$q_{t+1} = \sum_{t' < t} \Delta q_{t'}$$

Thus one comes up with the following discrete formula

$$\Delta q_{t'} = -\eta q_{t'}^{\kappa+1}. \quad (\text{A3})$$

Since $\kappa \geq 1$ and $1 > \eta > 0$, convergence towards null is obvious here. In addition since the right side of Equation (A3) is decreasing continuously it is obvious that it is converging fast than the corresponding solution of the differential equation for a function q_* :

$$\frac{dq_*}{dt} = -\eta q_*^{\kappa+1},$$

which is easily solvable to:

$$-\frac{1}{\kappa\eta}q_*^{-\kappa} = t + C,$$

where C is the integration constant. From this solution, by setting $q_0 = q_*(0)$ one can derive

$$q_*(t) = \left[\frac{\eta}{\kappa}t + q_0^{-\kappa}\right]^{-1/\kappa}.$$

Note that

$$\lim_{t \rightarrow \infty} q_* = 0.$$

Simple algebraic considerations show that $q_*(t)$ covers the sequence q_t . For a strict proof see Appendix B.

Thus,

$$q_{*,t} \geq q_t$$

for all $t > 0$ if

$$q_{*,0} = q_0$$

So, the sequences $q_{*,t}$ and q_t converge to zero.

Appendix B. Sequence $q_{*,t}$ Covers up Sequence q_t

In the following, identical definitions to Section 4 are used. One can start from the statement

$$-\frac{\eta}{\kappa} \geq -\eta. \quad (\text{B1})$$

Since $\eta \geq 0$ and $\kappa \geq 1$, it can easily be seen that the inequality is fulfilled for any combination of s , η and κ . One can now extend the numerator and denominator of the right side by $(\frac{\eta}{\kappa}t + C)$ and add also $(\frac{\eta}{\kappa}t + C)$ to both sides of the inequality. Here and in the following C is defined as

$$C = q_0^{-\kappa} \geq 1. \quad (\text{B2})$$

One obtains

$$\frac{\eta}{\kappa}t + C - \frac{\eta}{\kappa} \geq \left(\frac{\eta}{\kappa}t + C\right) - \left(\frac{\eta}{\kappa}t + C\right)\frac{\eta}{\frac{\eta}{\kappa}t + C}. \quad (\text{B3})$$

A rearrangement of the right side results in

$$\frac{\eta}{\kappa}t + C - \frac{\eta}{\kappa} \geq \left(\frac{\eta}{\kappa}t + C\right) \times \left(1 - \frac{\eta}{\frac{\eta}{\kappa}t + C}\right). \quad (\text{B4})$$

One can add $\frac{\eta}{\kappa}$ to both sides and obtain

$$\frac{\eta}{\kappa}t + C \geq \frac{s\eta}{\kappa} + \left(\frac{\eta}{\kappa}t + C\right) \times \left(1 - \frac{\eta}{\frac{s\eta}{\kappa}t + C}\right). \quad (\text{B5})$$

Now, one can use the fact that

$$1 \geq 1 - \frac{\eta}{\frac{\eta}{\kappa}t + C} \geq \left(1 - \frac{\eta}{\frac{\eta}{\kappa}t + C}\right)^\kappa$$

and

$$\frac{\eta}{\kappa} \geq \frac{\eta}{\kappa} \times \left(1 - \frac{\eta}{\frac{\eta}{\kappa}t + C}\right) \geq \frac{\eta}{\kappa} \times \left(1 - \frac{\eta}{\frac{\eta}{\kappa}t + C}\right)^\kappa$$

and thus rewrite inequality Equation (B5) as

$$\frac{\eta}{\kappa}t + C \geq \left(\frac{\eta}{\kappa}t + C + \frac{\eta}{\kappa}\right) \times \left(1 - \frac{\eta}{\frac{\eta}{\kappa}t + C}\right)$$

and finally arrive at

$$\left(\frac{\eta}{\kappa}t + C\right)^\kappa \geq \left(\frac{\eta}{\kappa}t + C + \frac{\eta}{\kappa}\right)^\kappa \times \left(1 - \frac{\eta}{\frac{\eta}{\kappa}t + C}\right)^\kappa.$$

One can multiply both sides by $\left(\frac{\eta}{\kappa}t + C + \frac{\eta}{\kappa}\right)^{-1} \times \left(\frac{\eta}{\kappa}t + C\right)^{-1}$. So,

$$\left(\frac{\eta}{\kappa}t + C + \frac{\eta}{\kappa}\right)^{-1} \geq \left(\frac{\eta}{\kappa}t + C\right)^{-1} \times \left(1 - \frac{\eta}{\frac{\eta}{\kappa}t + C}\right)^\kappa.$$

Taking the k th root on both sides one gets

$$\left(\frac{\eta}{\kappa}t + C + \frac{\eta}{\kappa}\right)^{-1/\kappa} \geq \left(\frac{\eta}{\kappa}t + C\right)^{-1/\kappa} \times \left(1 - \frac{\eta}{\frac{\eta}{\kappa}t + C}\right).$$

One can rearrange the inequality to:

$$\begin{aligned} \left(\frac{\eta}{\kappa}t + C + \frac{\eta}{\kappa}\right)^{-1/\kappa} &\geq \left(\frac{\eta}{\kappa}t + C\right)^{-1/\kappa} \times \left(1 - \eta \left(\frac{\eta}{\kappa}t + C\right)^{-1}\right) \\ \left(\frac{\eta}{\kappa}(t+1) + C\right)^{-1/\kappa} &\geq \left(\frac{\eta}{\kappa}t + C\right)^{-1/\kappa} \times \left(1 - \eta \left(\frac{\eta}{\kappa}t + C\right)^{-1/\kappa}\right)^\kappa \end{aligned} \tag{B6}$$

Using the definitions of $q_{*,t}$ and $q_{*,t+1}$, i.e.,

$$q^*(t) = \left[\frac{\eta}{\kappa}t + q_0^{-\kappa}\right]^{-1/\kappa},$$

one has finally

$$q_{*,t+1} \geq q_{*,t} \times (1 - \eta(q_{*,t})^\kappa). \tag{B7}$$

proven as a true statement. Thus,

$$q_{*,t} \geq q_t \geq d(\mathbf{y}_t, \mathbf{x}_t)$$

for all $t > 0$ if

$$q_{*,0} = q_0 = d(\mathbf{y}_0, \mathbf{x}_0).$$

Thus, the positive definite sequence $q_{*,t}$ covers $d(\mathbf{y}_t, \mathbf{x}_t)$.

Appendix C. Weak Contraction with the Present Transfer Function

In this appendix a test function of the form of Equation (26), is verified for the function of Equation (16) and hyperbolic tangent. Within this section we test the following values

$$\eta = \frac{1}{48 \times n^2}, \gamma = \frac{1}{2} \text{ and } \kappa = 2, \tag{C1}$$

for both transfer functions and square norm ($\|\cdot\|_2$), and n -neurons. In order to derive these values, one can start by considering linear responses $\|\mathbf{y}_{lin,t} - \mathbf{x}_{lin,t}\|$ and the final value $\|\mathbf{y}_t - \mathbf{x}_t\|$ within one single neuron as

$$\begin{aligned} \mathbf{y}_t &= \theta(\mathbf{y}_{lin,t}) \\ \text{and } \mathbf{x}_t &= \theta(\mathbf{x}_{lin,t}). \end{aligned} \quad (\text{C2})$$

We can define

$$y_{lin,t,i} = \Delta_i + \zeta_i, \quad x_{lin,t,i} = \zeta_i, \quad \text{and} \quad x_{t,i} - y_{t,i} = \delta_i \quad (\text{C3})$$

where i is the index of the particular hidden layer neuron. Note that

$$\|\mathbf{y}_{lin,t} - \mathbf{x}_{lin,t}\|^2 = \sum_i \Delta_i^2, \quad \|\mathbf{y}_t - \mathbf{x}_t\|^2 = \sum_i \delta_i^2.$$

Appendix C.1. One-Neuron Transfer

In this section considerations are restricted to the case in which one has only one neuron in the hidden layer. For the sake of simplicity, the subscript index i is left out in the following considerations.

Setting in the definitions of Equation (C3) into the square of Equation (25) and for a single neuron and for any ζ we get

$$\Delta^2 \phi_1(\Delta^2) \geq (\theta(\Delta + \zeta) - \theta(\zeta))^2.$$

Thus, it suffices to consider

$$\phi_1(\Delta^2) \geq \max_{\zeta} \omega(\Delta, \zeta), \quad (\text{C4})$$

where

$$\omega(\Delta, \zeta) = \left(\frac{\theta(\Delta + \zeta) - \theta(\zeta)}{\Delta} \right)^2$$

The \max_{ζ} can be found by basic analysis. Extremal points can be found as solutions of

$$\begin{aligned} \frac{\partial}{\partial \zeta} \omega &= \\ \frac{\partial}{\partial \zeta} \left[\left(\frac{\theta(\Delta + \zeta) - \theta(\zeta)}{\Delta} \right)^2 \right] &= \\ \frac{2(\theta(\Delta + \zeta) - \theta(\zeta)) \times (\dot{\theta}(\Delta + \zeta) - \dot{\theta}(\zeta))}{\Delta^2} &= 0. \end{aligned} \quad (\text{C5})$$

This can only be fulfilled if

$$\dot{\theta}(\Delta + \zeta) - \dot{\theta}(\zeta) = 0$$

Since θ' is an even function for both suggested transfer functions, one gets $\zeta = -z/2$ as the extremal point. Fundamental analysis shows that this point in both cases is a maximum. Thus, requiring

$$\phi_1(\Delta^2) \geq \left[\frac{2\theta(\Delta/2)}{\Delta} \right]^2 \quad (\text{C6})$$

also would satisfy Equation (C4).

Numerically, one can find parameters for ϕ_1 that are $\eta = 1/48$ and $\kappa = 2$ of as in Equation (C1).

For $\Delta^2 > \gamma$, it suffices to check if

$$\frac{4 \tanh^2(\Delta/2)}{\Delta^2} \leq 1 - \eta \gamma^\kappa$$

First the inequality is fulfilled for $\Delta^2 = \gamma$. Since the left side of the equation above is strictly decreasing fulfilled for all values $\Delta^2 > \gamma$.

Analogous considerations lead to the same parameters to cover up the transfer function from Equation (16).

Appendix C.2. Multi-Neuron Parameters

For several neurons one has to consider the variational problem of all possible combinations of values of Δ_i . One can start from the proven relation from the previous section,

$$\delta_i^2 \leq \Delta_i^2 \phi_1(\Delta_i^2)$$

Thus,

$$\sum_i \delta_i^2 \leq \sum_i (\Delta_i^2 \phi_1(\Delta_i^2)),$$

implying that

$$\sum_i (\Delta_i^2 \phi_1(\Delta_i^2)) \leq \sum_i \Delta_i^2 - \Delta_{max}^2 + \Delta_{max}^2 \phi_1(\Delta_{max}^2), \quad (C7)$$

where $\Delta_{max}^2 = \max_i \Delta_i^2$. The smallest possible Δ_{max}^2 is

$$\frac{\sum_i \Delta_i^2}{n},$$

Substituting into Equation (C7), we get

$$\begin{aligned} \sum_i \Delta_i^2 - \Delta_{max}^2 + \Delta_{max}^2 \phi_1(\Delta_{max}^2) &\leq \\ \sum_i \Delta_i^2 - \frac{\sum_i \Delta_i^2}{n} + \frac{\sum_i \Delta_i^2}{n} \phi_1\left(\frac{\sum_i \Delta_i^2}{n}\right) &= \\ (\sum_i \Delta_i^2) \phi_1\left(\frac{\sum_i \Delta_i^2}{n^2}\right). \end{aligned}$$

Thus, the inequality (which is equivalent to Equation (25)),

$$\sum_i \delta_i^2 \leq \left(\sum_i \Delta_i^2\right) \phi_k\left(\sum_i \Delta_i^2\right),$$

is fulfilled if ϕ_k is defined as

$$\phi_k(x) = \phi_1\left(\frac{x}{n^2}\right).$$

Thus, the parameters from Equation (C1) fulfil the inequality of Equation (25).

References

1. Werbos, P.J. Backpropagation through time: What it does and how to do it. *Proc. IEEE* **1990**, *78*, 1550–1560.
2. Lukoševičius, M.; Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* **2009**, *3*, 127–149.
3. Maass, W.; Natschläger, T.; Markram, H. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Comput.* **2002**, *14*, 2531–2560.
4. Jaeger, H. The “echo state” approach to analysing and training recurrent neural networks—With an erratum note. In *GMD German National Research Institute for Computer Science 2001*; GMD Report 148. Available online: <https://pdfs.semanticscholar.org/8430/c0b9afa478ae660398704b11dca1221cf22.pdf> (accessed on 22 December 2016).
5. Boedeker, J.; Obst, O.; Lizier, J.; Mayer, N.M.; Asada, M. Information processing in echo state networks at the edge of chaos. *Theory Biosci.* **2012**, *131*, 205–213.

6. Manjunath, G.; Jaeger, H. Echo state property linked to an input: Exploring a fundamental characteristic of recurrent neural networks. *Neural Comput.* **2013**, *25*, 671–696.
7. Mayer, N.M. Adaptive critical reservoirs with power law forgetting of unexpected input events. *Neural Comput.* **2015**, *27*, 1102–1119.
8. Ng, A. CS 229 Machine Learning, UC Stanford. URL. 2014. Available online: <http://cs229.stanford.edu/notes/cs229-notes1.pdf> (accessed on 22 December 2016).
9. Jaeger, H.; Maass, W.; Principe, J. Special issue on echo state networks and liquid state machines. *Neural Netw.* **2007**, *20*, 287–289.
10. Schrauwen, B.; Verstraeten, D.; Van Campenhout, J. An overview of reservoir computing: Theory, applications and implementations. In Proceedings of the 15th European Symposium on Artificial Neural Networks, Bruges, Belgium, 25–27 April 2007.
11. Jaeger, H. Adaptive nonlinear system identification with echo state networks. In Proceedings of the Neural Information Processing Systems: Natural and Synthetic, Vancouver, BC, Canada, 9–14 December 2002.
12. Wainrib, G.; Galtier, M.N. A local Echo State Property through the largest Lyapunov exponent. *Neural Netw.* **2016**, *76*, 39–45.
13. Buehner, M.R.; Young, P. A Tighter Bound for the Echo State Property. *IEEE Trans. Neural Netw.* **2006**, *17*, 820–824.
14. Yildiz, I.B.; Jaeger, H.; Kiebel, S.J. Re-visiting the echo state property. *Neural Netw.* **2012**, *35*, 1–20.
15. Appeltant, L.; Soriano, M.C.; Van der Sande, G.; Danckaert, J.; Massar, S.; Dambre, J.; Schrauwen, B.; Mirasso, C.R.; Fischer, I. Information processing using a single dynamical node as complex system. *Nat. Commun.* **2011**, *2*, 468.
16. Ortín, S.; Soriano, M.C.; Pesquera, L.; Brunner, D.; San-Martín, D.L.; Fischer, I.; Mirasso, C.R.; Gutiérrez, J.M. A unified framework for reservoir computing and extreme learning machines based on a single time-delayed neuron. *Sci. Rep.* **2015**, *5*, 14945.
17. Natschläger, T.; Bertschinger, N.; Legenstein, R.A. At the edge of chaos: Real-time computations and self-organized criticality in recurrent neural networks. In Proceedings of the Advances in Neural Information Processing Systems 17, Bonn, Germany, 7–11 August 2005.
18. Legenstein, R.A.; Maass, W. Edge of chaos and prediction of computational performance for neural circuit models. *Neural Netw.* **2007**, *20*, 323–334.
19. Mayer, N.M. Critical Echo State Networks that Anticipate Input Using Adaptive Transfer Functions. *arXiv* 2016, arXiv:1411.6757.
20. Rubin, D.C.; Wenzel, A.E. One hundred years of forgetting: A quantitative description of retention. *Psychol. Rev.* **1996**, *103*, 734.
21. Kahana, M.J.; Adler, M. *Note on the Power Law of Forgetting*; University of Pennsylvania: Philadelphia, PA, USA, 2002.



© 2016 by the author; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).