Memory versus Non-Linearity in Reservoirs

David Verstraeten, Joni Dambre, Xavier Dutoit, Benjamin Schrauwen

Abstract— **Reservoir Computing (RC) is increasingly being used as a conceptually simple yet powerful method for using the temporal processing of recurrent neural networks (RNN). However, because fundamental insight in the exact functionality of the reservoir is as yet still lacking, in practice there is still a lot of manual parameter tweaking or brute-force searching involved in optimizing these systems. In this contribution we aim to enhance the insights into reservoir operation, by experimentally studying the interplay of the two crucial reservoir properties, memory and non-linear mapping. For this, we introduce a novel metric which measures the deviation of the reservoir from a linear regime and use it to define different regions of dynamical behaviour. Next, we study the relationship of two important reservoir parameters, input scaling and spectral radius, on two properties of an artificial task, namely memory and non-linearity.**

I. Introduction

Reservoir Computing is an appealing method in the field of neural network research, due to the combination of its conceptual simplicity and its good performance on a variety of tasks. There are several viewpoints from which to describe the functionality of the reservoir. For instance, it bears some similarity to a kernel map in the sense that a reservoir, too, performs a mapping of the input into a much higher-dimensional space which, in the case of classification tasks, is known to increase the probability of linear separability of the data [2]. Alternatively, the reservoir can be said to boost the computational power of the simple memoryless linear readout by providing both a non-linear mapping and a fading memory of the inputs [11]. This combined temporal processing and non-linear mapping is required for many interesting or challenging real-world tasks.

Reservoir Computing allows the user to incorporate a certain amount of intuition into the design of the system. Because of its generically defined architecture and broad range of possible extensions, RC systems can be easily adjusted and extended in various directions, including specialized nodes [17], network topologies [4] or reservoir adaptation rules [15]. The review paper [10] gives a very complete overview of the different RC variants described in literature.

The fundamental idea is easy enough: take a random recurrent network of simple nodes (such as sigmoid neurons) called the reservoir, globally scale its weights to impose a desired dynamical regime and use the response of all neurons in the reservoir to train a linear readout using any of the available linear regression algorithms to approximate the desired output. Still, even in such a simple setting, quite a number of global parameters influence the reservoir's overall dynamic regime. Many of these can be tuned to optimize the reservoir dynamics to a given task. Unfortunately, an underlying theoretical framework or a principled approach for designing good reservoirs is still lacking. Hence, in practice one of the major benefits of RC (its short training times compared to other recurrent NN learning rules) is somewhat diminished by extensive parameter searches needed to find optimal reservoir settings. It is no surprise then, that the search for directed optimization methods or a more complete mathematical theory of the internal operation of reservoirs is a very active topic of research.

Once the neuron type has been chosen, the global reservoir parameters can be roughly divided into several categories: scaling parameters, temporal parameters and topological parameters. Of the topological parameters, such as the average neuron fan-in, it has been known for some time from experimental work that they do not greatly affect reservoir performance for analogue neurons [18]. In [1] a theoretical framework was developed to consolidate this finding. Several temporal parameters, such as the leak rate for leaky integrator neurons [6] or the relation between the reservoir reaction time and the relevant time scales of the input signal [16], have been found to drastically affect reservoir performance for a given task, but a systematic analysis of their impact is still missing.

In this work, we focus on the scaling parameters for the connection weights in reservoirs with hyperbolic tangent neurons. Together with the neuron bias, they determine in which parts of their nonlinear transfer curves the neurons operate. For small input signals, the transfer curve of these neurons is approximately linear, whereas for large input signals, it is very nonlinear. In the linear regime, for small input signals and small bias, the reservoir properties approximate those of linear reservoirs, for which some analytical results have been published [5, 20]. In particular, it has been found that its memory capacity can be tuned between very good short term memory and (lower quality) long term memory by rescaling the reservoir connection matrix [3]. Hence, this scaling parameter can be used to optimally tune the reservoir to the memory requirements of a given task.

However, most challenging real-world tasks require both memory and a non-linear mapping. To achieve sufficient *non-linear richness* in a reservoir, at least part of the neurons have to be tuned away from their linear regime by rescaling the input or the reservoir connection matrix or by setting an appropriate bias level. It is generally assumed that increasing nonlinearity degrades memory capacity, but very little is

David Verstraeten, Joni Dambre and Benjamin Schrauwen are with the Department of Electronics and Information Systems (ELIS), Ghent University, Sint-Pietersnieuwstraat 41, 9000 Ghent, Belgium. E-mail: david.verstraeten@ugent.be, joni.dambre@ugent.be, benjamin.schrauwen@ugent.be.

Xavier Dutoit is with the Department of Mechanical Engineering, Catholic University Leuven, Celestijnenlaan 300b - bus 2420, 3001 Leuven, Belgium. E-mail: xavier.dutoit@mech.kuleuven.be.

978-1-4244-8126-2/10/$26.00 ©2010 IEEE

known about the interplay between them and the extent to which this can be explored by tuning the scaling parameters.

In this paper, we will investigate if certain dynamical regions can be discerned and the way these regions relate to the main reservoir parameters. In Section III we use on the one hand a well known measure of stability for dynamical systems, the maximal local Lyapunov exponent, and on the other hand we introduce a novel measure which quantifies the deviation of a reservoir from a linear system. We investigate the relation between these measures and reservoir settings and use the results to identify certain regions of operation with different properties with regards to the non-linear mapping. Next, in Section IV, we study the relation between these found dynamical regimes and the properties of an actual (artificial) task which offers accurate control over the amount of non-linearity and memory required from the reservoir. We discuss the trade-off made between memory and non-linear requirements of the task. Finally, we summarize and draw conclusions in Section V.

II. Definitions

We will use standard ESNs [9] throughout this contribution, but the results presented here can be transferred to most other reservoir types. For the sake of completeness, we include the equations governing the system update. At a given time k, the (possibly multidimensional) input to the reservoir is denoted $\mathbf{u}[k]$, the vector containing the output of the neuron's activation function as $\mathbf{x}[k]$ and the desired output as $\mathbf{y}[k]$. The output generated by the trained RC system is denoted $\hat{\mathbf{y}}[k]$. The weights projecting the input into the reservoir are $\mathbf{W}_{in}$ and the internal reservoir weights are $\mathbf{W}_{res}$. In this contribution, both weight matrices are full and their elements are drawn from a Gaussian distribution with unit variance. The input weight matrix is then rescaled with an input scaling factor ι. As usual for ESNs, the reservoir weight matrix is rescaled to a desired spectral radius ρ [7]. Every neuron receives a bias, represented by the constant vector $\mathbf{b}$**,** that is also randomly drawn from a Gaussian distribution and scaled by a factor β. Thus, the global parameters for the reservoir under consideration here are the three scaling factors for input, reservoir and bias: $[\iota,\rho,\beta]$.

The network is initialized to zero ($\mathbf{x}[0] = \mathbf{0}$), and the response of the network to the input signal $\mathbf{u}_k,\ k = 1 \dots T$ is computed using the following update equation:

$$\mathbf{x}[k+1] = f\left(\mathbf{W}_{res}\mathbf{x}[k] + \mathbf{W}_{in}\mathbf{u}[k] + \mathbf{b}\right).$$

The response of the reservoir to the whole train set is collected in this way. The matrix $\mathbf{A}$ is constructed as the concatenation of the reservoir states $\mathbf{x}$ and inputs $\mathbf{u}$: $\mathbf{A} = [\mathbf{x}; \mathbf{u}]$**.** The weights of the linear readout $\mathbf{W}_{out}$ are then computed by minimizing the squared error on the train outputs:

$$\mathbf{W}_{out} = \min_{\mathbf{W}} \|\mathbf{W}\mathbf{A} - \mathbf{y}\|^2,$$

which can be done in a single step using the Moore-Penrose pseudo-inverse:

$$\mathbf{W}_{out} = \mathbf{A}^{\dagger}\mathbf{Y} = \left(\mathbf{A}^{\mathbf{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathbf{T}}\mathbf{Y},$$

where $\mathbf{X}$ and $\mathbf{Y}$ denote the concatenation of all $\mathbf{x}$ and $\mathbf{y}$ vectors over time. The output of the RC system is then obtained as:

$$\hat{\mathbf{y}}[k] = \mathbf{W}_{out}\left[\mathbf{x}[k]; \mathbf{u}[k]\right]$$

III. Quantifying reservoir nonlinearity

With this work, we aim to chart the effect of the three parameters ι, β and ρ on the dynamical properties of the reservoir and in particular on its nonlinearity. For this purpose, this section introduces a measure that quantifies the overall reservoir nonlinearity, based on frequency analysis of the reservoir states. Although we do not explicitly study memory in this section, we do relate our observations to a second measure that quantifies the stability or excitability of the reservoir. This measure also identifies the so-called edge-of-stability, a dynamic regime that is usually required for obtaining long-term input-driven memory in a dynamic system. We study the impact on both measures of the reservoir parameters ι, β and ρ.

A. Deviation from linearity

The first measure, which we will call deviation from linearity in the frequency domain and denote by δ_ϕ, is inspired by the fact that that the frequency spectrum of a linear time-invariant system does not show additional frequencies when fed with a pure single frequency input (a sine wave). Non-linear systems do however introduce harmonics. Therefore, it makes sense to feed a pure sine wave into the reservoir and to measure the amount of energy contained in the reservoir states at that frequency. We can then define the deviation from linearity based on the ratio between the energy in the input frequency, and the energy contained all other frequencies. In practice, we compute this measure as follows:

- The reservoir is fed with a sine wave of a certain carrier frequency f_c.
- The Fast Fourier Transform (FFT) of the reservoir states is computed and averaged over all neurons. The total energy of the averaged FFT minus the DC component is defined as E_{tot}.
- The energy of the FFT at f_c is E_c, and the deviation from non-linearity is defined as $\delta_\phi = 1 - \frac{E_c}{E_{tot}}$.
- This experiment is repeated for 100 different carrier frequencies f_c, linearly spaced between 0.01 and 0.5 cycles per timestep and averaged over the different carrier frequencies.

We note that δ_ϕ does not make any assumptions about the properties of the reservoir, which means that it can be applied to other reservoir types than the standard ESNs.

We need to point out here that the measure δ_ϕ merely expresses a certain relation between input and output spectra of a reservoir. We will see that this does not necessarily

mean that this relation is an expression of an underlying causal relationship. Especially high values of δ_ϕ can also be obtained for spectra that are essentially unrelated.

B. Lyapunov stability

A second measure of the reservoir's dynamical regime is the maximal Lyapunov exponent. Lyapunov exponents measure the exponential deviation of a dynamical system from its trajectory when it is perturbed. One can define temporally local approximations of these exponents, called Local Lyapunov Exponents (LLE) [19, 13]. The LLEs can be computed by considering the Jacobian J of the reservoir. In this paper we consider standard neurons with a $\tanh$ activation function, for which the Jacobian can be computed analitically. The i,jth element of this Jacobian at a time k is given by:

$$J_{i,j}[k] = \begin{bmatrix} 1-x_1^2[k] & 0 & \dots & 0 \\ 0 & 1-x_2^2[k] & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1-x_n^2[k] \end{bmatrix} * \mathbf{W}_{res},$$

where $x_i[k]$ is the output of the ith neuron at time k. The Local Lyapunov exponents can then be obtained based on the eigenvalue spectrum of the Jacobian:

$$LLE_i = log\left(\prod_t^T |\lambda_i[k]|^{1/T}\right),$$

where $\lambda_i[k]$ is the ith eigenvalue of the Jacobian at time k, and T is the number of timesteps in the trajectory. In this case, the input is a signal consisting of uncorrelated (white) samples from a uniform distribution over $[-0.8, 0.8]$.

There is a whole spectrum of Lyapunov exponents (one for each distinct eigenvalue of the Jacobian), each measuring the deviation of the system in a certain orthogonal direction in state space - a direction given by the corresponding eigenvector. However, in general one is usually interested in the maximal exponent, LLE_{max}. If this exponent is larger than zero, this means that the system will deviate exponentially from its trajectory in at least one direction, which means the system is Lyapunov unstable. For a reservoir, this means that its internal dynamics become dominant and the system fails to perform any useful mapping of the input signal.[1]

C. Visually combining both measures

We now visualise the combined parameter impact on both measures of the dynamical regime into a single plot (Figure 1). On the one hand, we plot a contour line showing the values of (ι,ρ) where δ_ϕ equals a given value (in this case 0.8), and this for a given value of the bias β (in this case 0). We range over ι and ρ in a logarithmic fashion to clearly show the trends. On the other hand, we also show the contour line where $LLE_{max} = 0$ on the same plots, corresponding to the edge of Lyapunov stability[2] for the reservoir (in the limit case where it is not driven by an external signal). It indicates the boundary of useful reservoir operation.

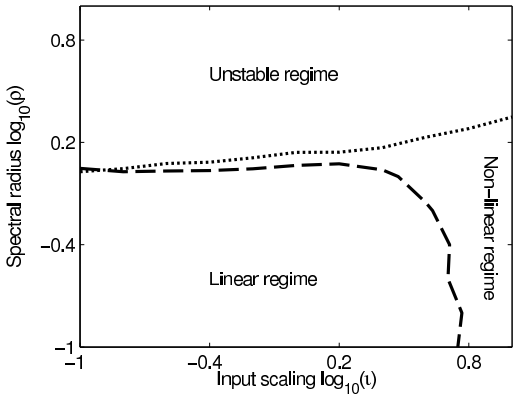


Fig. 1: Different dynamical regimes for reservoirs.

In this figure, we can define three different dynamical regions of interest. For large values of ρ, the reservoir operates in the unstable regime, where the reservoir is highly excitable and does not respond to changes in the input: the internal reservoir dynamics drown out all external information. For very large values of ρ the reservoir can even become chaotic.

For smaller values of ρ we can discern two additional areas, depending on the value of the input scaling. For small input scaling factors, the reservoir operates in the linear regime: the input is mapped more or less linearly into the reservoir state, i.e., it is only slightly distorted. This regime corresponds to the region of maximal memory. From [3], we know that, within the linear regime, the type of memory (short-term vs. long term) can be tuned by changing the spectral radius: for long term memory, the dynamic regime must be closer to the edge of stability.

For larger values of the input scaling, we can identify a non-linear regime. Here, the input is fed into the reservoir with a high energy, causing it to undergo substantial non-linear transormation. While this has a detrimental effect on the memory of the reservoir, as we will see later this non-linear mapping can be beneficial for some types of tasks.

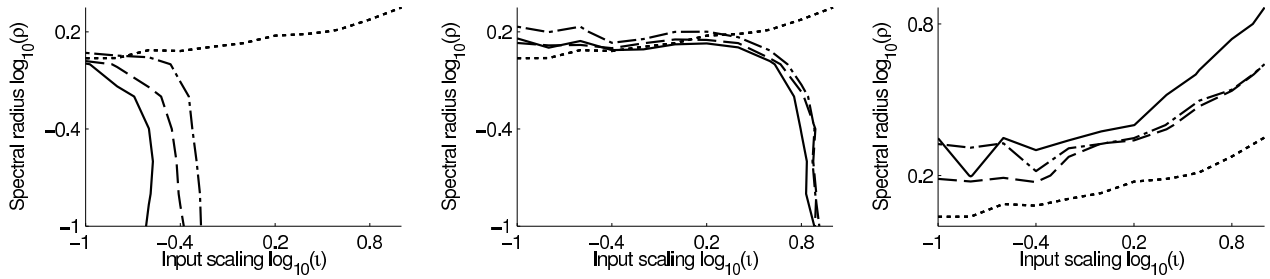
D. Impact of reservoir parameters

We now use the type of plot introduced in the previous section to visualize the combined parameter impact on both measures of the dynamical regime (Figure 2). Each individual plot now shows contour lines for a single value of δ_ϕ, but for several values of the bias β. The line for $LLE_{max} = 0$ is still plotted for zero bias only, since we found it to show very little dependence on β. This plot was made for $\delta_\phi = 0.6$ (Fig. 2a), $\delta_\phi = 0.8$ (Fig. 2b) and $\delta_\phi = 0.95$ (Fig. 2c). From these plots, some observations can be made.

Let us first focus on the significance of the line where $LLE_{max} = 0$. Roughly speaking, parameter combinations

[1]Because the LLE is computed based on the Jacobian of the system, the measure can in principle be computed for any system which has a continously differentiable mapping, which means that it can be applied to other reservoirs beyond the standard ESNs.

[2]Often erroneously called the edge of chaos. In fact there are a multitude of non-chaotic dynamical regimes beyond this line, such as limit cycles, self-oscillations and others.



(a) $\delta_\phi = 0.6$ (b) $\delta_\phi = 0.8$ (c) $\delta_\phi = 0.95$

Fig. 2: Contour lines for different values of the deviation from linearity δ_ϕ. In every plot, the full line is a contour for $\log_{10}(\beta) = -1$, the dashed line for $\log_{10}(\beta) = -0.2$ and the dash-dot line for $\log_{10}(\beta) = 1$. The dotted line is the contour for $LLE_{max} = 0$, indicating the edge of stability. This line is the same in all three plots (note the different scaling of the Y-axis for sub-plot (c)).

(ι,ρ) above this line yield reservoirs that barely respond to the input signal applied and are therefore not useful. Whenever a δ_ϕ-isoline falls significantly above this boundary, the observed value of the nonlinearity measure is no longer the result of an actual input-output relationship. This occurs in Figure 2c, indicating in fact that a δ_ϕ that is as high as 0.95 *and* still reflecting a useful highly nonlinear mapping of the input signal can not be attained with the bias values shown. The fact that the isolines for $\beta = \log_{10}(-1)$ and $\beta = \log_{10}(-0.2)$ are very close together, indicates that this level of nonlinearity can probably not be attained at all in the class of reservoirs studied in this paper. This observation suggests that there is a fundamental limit to the degree of nonlinearity a given parameterized reservoir class can provide.

Next, we consider Figures 2a and 2b. In neither of these figures, the isolines are systematically or very significantly above the stability boundary[3]. Clearly, input-output mappings with nonlinearity levels characterized by δ_ϕ-values at least up to 0.8 can be realised by the class of reservoirs under consideration.

In both figures, and for each individual isoline, we find the three dynamic regions identified in Section III-C. For very linear regimes, we find an inversely proportional relation between input and bias scaling. This is because a higher bias shifts the operating point towards more non-linear regions, which needs to be compensated by a larger input scaling in order to still have part of the input signal end up in the linear region of the neuron's activation function.

In the stable regions, and for very linear reservoirs (small ι and β), all three parameters affect nonlinearity. Moving away from the linear regime, the effect of the tuning parameters differs depending on the distance to the stability boundary.

Sufficiently far from this boundary, the nonlinearity can be tuned by both the input scaling and the bias. The spectral radius has very little impact in this region. The impact of bias changes is more pronounced for lower nonlinearity levels (and hence for lower values of ι). For higher nonlinearity levels, obtained by higher values of ι, the impact of β decreases. The same is true for the impact of the input scaling (not shown in these figures). This saturation of the achievable nonlinearity can be intuitively explained as follows. When driving the neurons in a very nonlinear regime, either by drastically shifting their transfer curves (high β) or by driving them with high input amplitudes (ι), not much additional nonlinearity is to be gained by further increasing either parameter. It can be expected that the trade-off between both lies in the type of nonlinearity that is required. For zero-mean input signals, tanh-neurons always produce a symmetric nonlinearity. Hence, increasing the amplitude of the neuron input signals will always lead to an increased symmetric nonlinearity. This can be achieved by increasing the input scaling or, in the regions where it has an effect on nonlinearity, by increasing the spectral radius. Adding a neuron bias increases the nonlinearity in an asymmetric way.

For close-to-linear reservoirs, we find that the reservoir nonlinearity is more sensitive to the neuron bias than for more nonlinear ones. This indicates that small levels of nonlinearity can be obtained either by increasing the input scaling or by increasing the bias scaling. For a reservoir that is already very nonlinear, increasing the neuron bias barely affects the dynamics.

Closer to the stability boundary, the situation is reversed. Here, the impact of input scaling and bias diminishes, especially for higher levels of nonlinearity. From the comparison of Figures 2a and 2b we can conclude that in this region the spectral radius ρ does affect the nonlinearity. Hence, for fixed input scaling, δ_ϕ can still slightly be tuned by changing ρ. However, as the spectral radius also controls memory, a trade-off is to be expected in this region.

The discussion above focuses on the achievable level of nonlinearity only. The performance of a reservoir for a given task also greatly depends on the reservoir's memory properties. Increasing nonlinearity is known to reduce memory, so a trade-off will have to be made. Here, memory will

[3]Note that the line for $LLE_{max} = 0$ in the figures only presents an estimate of the actual stability boundary, so some uncertainty margin should be considered here.

always dominate. A reservoir that reflects the necessary input history too linearly will always perform better than a nonlinear reservoir where this history is lost. Hence, we can expect that the margin for tuning nonlinearity will decrease as memory requirements become more dominant. This trade-off is investigated in the next section.

IV. Linking the reservoir dynamics to task properties

In order to study the interplay between non-linearity of the mapping and memory in the reservoir, we introduce a task which allows accurate control over both the required memory and non-linear mapping. The input signal $u[k]$ consists of uncorrelated values from a uniform distribution over the interval $[-0.8, 0.8]$. The advantage of this signal is that - because of its lack of temporal correlation - in theory all frequencies are present with equal energy: the signal is uniform white noise. From this input signal, the task is then to reconstruct delayed nonlinear versions of the input, as follows:

$$y_{d,p}[k] = sign(r[k-d])abs(r[k-d])^p,$$

where r represents the product of two delayed successive inputs: $r[k-d] = u[k-d]u[k-d-1]$. The sign and absolute values are introduced to assure a symmetric output even in the case of even powers (which would otherwise be strictly positive). Asymmetric signals are more difficult to extract from a reservoir with symmetric inputs, a symmetric nonlinearity and a symmetric input and reservoir weight distribution, so this would introduce an unnecessary difference between uneven and even powers.

This task can be seen as a continuous extension of the delayed XOR-task used in [14]. There, a stochastic stream of bits was used as inputs, and the task was to recall the binary XOR of three delayed bits. The XOR function is nonlinear: in neural networks research it is well known as the task which showed the limitations of the linear perceptron [12]. When transferred to the continuous domain $[-1, 1]$, the same type of non-linear mapping of two variables can be obtained by their product (see table I). By taking the power p of this product, this mapping becomes even more nonlinear (see Fig. 3).

	0	1
0	0	1
1	1	0

(a) Truth table for binary XOR

	-1	1
-1	1	-1
1	-1	1

(b) The product of two symmetric continuous variables

TABLE I: Extending the XOR function of two binary variables to continuous values.

This task allows us to study independently the interplay of the two main requirements of reservoirs: memory and non-linear mapping. The task memory is determined by the delay d of the output, and the degree of task non-linearity is determined by the parameter p.

For our experiments, we trained several readouts simultaneously on each of the outputs $y_{d,p}$. We considered delays $d = 1, \ldots, 15$ and powers $p = 1, \ldots, 10$, yielding a total of 150 output signals. For each experiment, ten different input streams of 1000 timesteps each and their corresponding outputs were generated.

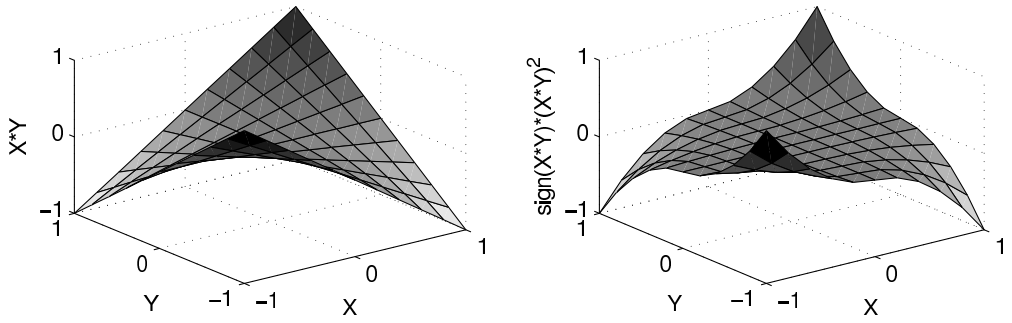
The reservoirs were constructed as follows: the reservoir and input weight matrix and the bias vector are drawn from a gaussian distribution with unit standard deviation. The input weight matrix W_{in} and bias vector $\mathbf{b}$ are rescaled by a constant parameter ι and β respectively, and the reservoir weight matrix W_{res} is globally rescaled so that its spectral radius (its largest absolute value) is equal to the parameter ρ, as is standard practice in RC. The readout weights W_{out} are trained using pseudo-inverse, as described above. Because the input to the reservoir is essentially a noise signal, overfitting is unlikely so no regularization was used. Every experiment was repeated ten times for the same parameter settings to even out variations in the random creation of the reservoirs, and the results were averaged over the different runs. The measure of performance for a single output $\hat{y}_{d,p}$ is the Normalized Root Mean Squared Error:

$$NRMSE_{d,p} = \left\langle \sqrt{\frac{(\hat{y}_{d,p}[k] - y_{d,p}[k])^2}{\sigma^2_{y,d,p}}} \right\rangle_k,$$

where $\sigma^2_{y,d,p}$ is the variance of the desired output signal and $\langle\rangle_k$ denotes an average over time k. The results were then averaged using 5-fold cross-validation on the dataset. We used 100 neurons in the reservoir.

We studied the effects that the main reservoir parameters (input scaling ι, spectral radius ρ and bias β) have with regards to the required memory and non-linearity of the task. We ranged over both ι and ρ, and β and ρ jointly (keeping the remaining parameter constant at $\iota = 1$ and $\beta = 0.1$), because the effects of the parameters are not independent. Since we want to investigate the influence of these parameters w.r.t. the task properties, we looked at the optimal parameter settings for different values of the delay and power in the task. In other words, for a given delay d and power p, we looked up which input scaling ι, bias scaling β and spectral radius ρ gave the lowest NRMSE. The results for ranging over β and ρ are shown in Figure 4, while those for ranging over ι and ρ are shown in Figure 5.

Figures 4c and 5c illustrate the overall optimal performance of our parameterized reservoir class for each combination of d and power p. Whereas the minimal delay task without additional power (the *baseline* task, $d = 0,\ p = 1$) is well within the range of the studied reservoir, the task rapidly becomes harder to solve as either its nonlinearity or its delay are increased. The fact that the baseline task is already quite nonlinear (a product of two input values), is reflected in the fact that the optimal values for both β and ι (Figures 4b and 5b, respectively) are considerably above zero.



(a) Simple product of two variables.

(b) Square of the product of two variables.

Fig. 3: The product of two values in [-1, 1] is a non-linear mapping. Taking the power of this product increases the non-linearity.

We first note that the optimal spectral radius plot is different when ranging over ρ and β jointly (Fig. 4a) than when ranging over ρ and ι jointly (Fig. 5a). This confirms our earlier claim that the three parameters are interdependent and their effects on task performance is not mutually independent.

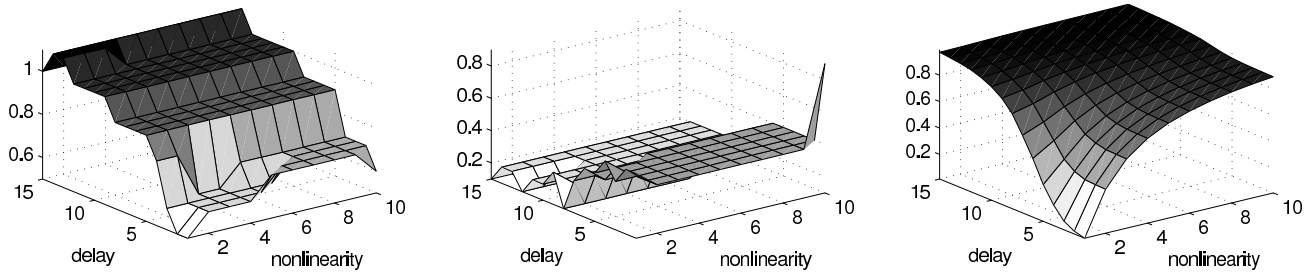
The plots for the optimal spectral radius (Figures 4a and 5a) confirm what was already known, namely that for longer task memory, the spectral radius should become larger - the dynamics of the reservoir are then better able to remember inputs from longer ago [3]. However, as Figures 4c and 5c show, the quality of the recollection of the past signals deteriorates rapidly for longer delays. Figures 4b and 5b also confirm that, as more memory is required, the optimal values of the neuron bias and the input scaling systematically decrease. This reflects the fact that the longer-term memory is optimal for linear reservoirs [8] and therefore any nonlinearity introduced by both parameters decreases the reservoir's memory performance. Hence, as memory becomes more important, ι and β settle at the maximal allowable level within the task's memory requirements. For this task, this is always lower than the optimal values for realising the required nonlinearity, which can be observed in both plots on the zero-delay lines. The optimal values therefore become independent of the actual task nonlinearity. In summary, our results for the delayed tasks show that, even for relatively small delays the memory requirements are dominant, leaving no freedom in parameter space for optimizing nonlinearity.

We do notice a dependence of optimal input bias and spectral radius on nonlinearity for very short delays (1-2 timesteps) in the plots for constant input scaling (Figures 4a and 4b). Here we see the trade-off between different types of nonlinearity. From the baseline task, we see that a product-type nonlinearity requires some neuron bias. This was also confirmed experimentally by the fact that for zero bias, all $(\rho,\beta)-$parameter combinations yielded very poor results, even for the baseline task (results not shown). Remember that, for this sweep, the input scaling was set to a relatively low value, a region where the spectral radius also has a significant effect on reservoir nonlinearity. It appears that, for low input scaling, the memory requirements for this task are easily fulfilled for small delays, leaving both the spectral radius and the neuron bias available to tune nonlinearity. Since both degrade the short-term memory of the task, a trade-off must be made to safeguard the memory that is still necessary for the task. The results illustrate this trade-off between the need for asymmetric nonlinearity, required to generate the product terms, and the need for increased symmetric nonlinearity as the power p increases. The former is obtained by increasing the bias, whereas the latter is obtained by increasing the spectral radius.

In contrast, hardly any dependence on task nonlinearity can be found in the plots for constant bias, not even for very short delays (Figures 5a and 5b). Here several explanations are possible. First note that, in the near-linear regime, both spectral radius and input scaling can tune symmetric nonlinearity only. Therefore, the trade-off between two types of nonlinearity discussed in the previous paragraph is no longer relevant here. An explanation for the fact that neither parameter significantly changes for increased nonlinearity indicates that in these plots, even for $d = 0$, the optimal solution is already restricted by the memory requirements. Remember that even in the baseline task, the output depends on two consecutive input values. This claim is supported by the fact that, for zero delay, optimal performances for all nonlinearity levels, achieved for input scaling values above 1.2, are better than in the constant input scaling case, where $\iota = 1$. In other words, within the fullfilment of memory requirements, all remaining freedom was used up by increasing the input scaling. Since this yields better performance than tuning the other two parameters, it is best for this task to maximally increase the input scaling while still providing a minimal amount of bias for generating the product terms and, above all, fulfilling the memory requirements.

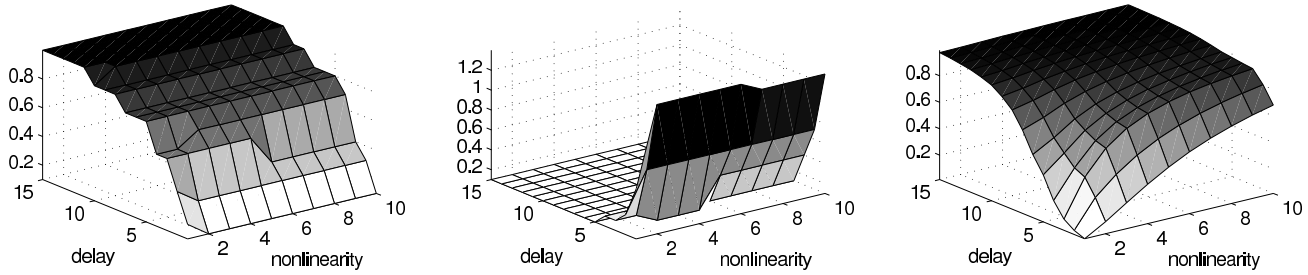
V. Conclusions

For many tasks, a reservoir needs to be able to perform a rich nonlinear spatiotemporal mapping of its input signals. To achieve this, it must be sufficiently nonlinear and have appropriate memory.



(a) Optimal ρ (b) Optimal β (c) Optimal NRMSE

Fig. 4: Optimal ρ and β values for different task delays d and non-linearities p.



(a) Optimal ρ (b) Optimal ι (c) Optimal NRMSE

Fig. 5: Optimal ρ and ι values for different task delays d and non-linearities p.

In the first part of this contribution we have presented an experimental investigation of the trade-off between non-linearity and stability of reservoir responses for classical tanh ESNs, parameterized by the three main reservoir scaling parameters, bias, input scaling and spectral radius. To quantify the nonlinearity of the reservoir outputs, we have used a measure of the deviation of the reservoir dynamics from a linear system based on the frequency spectrum. To monitor the extent to which the resulting nonlinearity levels still reflect a reservoir's response to the applied input signals, we also measured the maximal local Lyapunov exponent for the same reservoir parameter settings. We identified three different dynamical regimes of interest, a linear regime, a non-linear stable regime and an unstable regime, and found that input-output-mappings with very high levels of the presented nonlinearity measure are not achievable with the reservoir class (and reservoir size) considered in this work. For the stable regimes, we again identified three regions with different types of dependencies on the reservoir parameters. In the very linear region, all three parameters have a significant impact on nonlinearity. The neuron bias yields asymmetric nonlinearity, whereas the other two regulate symmetric nonlinearity. When the system is sufficiently stable and at least somewhat nonlinear, the spectral radius no longer affects nonlinearity. In contrast, close to the stability boundary, spectral radius becomes the only parameter that still has an impact on the nonlinearity measure, but in this region, the reservoir dynamics already become less sensitive to the input values.

In the second part of this work, we investigated the interplay between the non-linearity and memory requirements of a task and the way these can be combined into optimally tuned reservoirs. For this purpose, we have proposed a new task that allows direct control over the required memory and non-linearity and is a direct extension of two classical benchmark tasks in the RC literature, namely the memory capacity task and the delayed XOR-task. Our results show that, for this task and reservoir size, overall reservoir performance is mostly dominated by the memory requirements. Only within the margins of fulfilling these requirements, some freedom remains for optimizing the system's nonlinearity. The more important long-term memory becomes for the task, the more nonlinearity is sacrificed and the optimal reservoir is steered towards the linear regime. Whenever some margin for optimization exists, we were able to link the observed optimal parameter settings the the observations made in the first part of this work.

Reservoir Computing is sometimes criticized for its black-box nature, but the results of this contribution will allow the users and designers of RC to make more educated adjustments and to arrive at the optimal settings quicker. Future work will be directed towards a more quantitative way to describe task requirements and their relation to optimal reservoir settings.

ACKNOWLEDGEMENTS

This research was partially supported by the Photonics@be Interuniversity Attraction Poles program (IAP 6/10), initiated by the Belgian State, Prime Minister's Services, Science

Policy Office, and the GOA Project HomeMATE funded by the Ghent university Special Research Fund.

REFERENCES

[1] Lars Buesing, Benjamin Schrauwen, and Robert Legenstein. Connectivity, dynamics, and memory in reservoir computing with binary and analog neurons. *Neural Computation*, 22(5):1272–1311, 2010.

[2] T.M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, 14(3):326–334, 1965.

[3] S. Ganguli, D. Huh, and H. Sompolinsky. Memory traces in dynamical systems. *Proceedings of the National Academy of Sciences*, 105(48):18970, 2008.

[4] M.A. Hajnal and A. Lorincz. Critical Echo State Networks. *Lecture notes in Computer Science*, 4131:658, 2006.

[5] Michiel Hermans and Benjamin Schrauwen. Memory in linear recurrent neural networks in continuous time. *Neural Networks*, 2010.

[6] H. Jaeger, M. Lukosevicius, and D. Popovici. Optimization and applications of echo state networks with leaky integrator neurons. *Neural Networks*, 20:335–352, 2007.

[7] Herbert Jaeger. The "echo state" approach to analysing and training recurrent neural networks. Technical Report GMD Report 148, German National Research Center for Information Technology, 2001.

[8] Herbert Jaeger. Short term memory in echo state networks. Technical Report GMD Report 152, German National Research Center for Information Technology, 2001.

[9] Herbert Jaeger and Harald Haas. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless telecommunication. *Science*, 308:78–80, April 2 2004.

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

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

[12] M. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, MA, 1969.

[13] M. C. Ozturk, D. Xu, and J. C. Principe. Analysis and design of echo state networks. *Neural Computation*, 19:111–138, 2006.

[14] B. Schrauwen, L. Busing, and R. Legenstein. On Computational Power and the Order-Chaos Phase Transition in Reservoir Computing. In *Proceedings of NIPS*, 2008.

[15] B. Schrauwen, M. Warderman, D. Verstraeten, J. J. Steil, and D. Stroobandt. Improving reservoirs using intrinsic plasticity. *Neurocomputing*, 71:1159–1171, 2008.

[16] Benjamin Schrauwen, Jeroen Defour, David Verstraeten, and Jan Van Campenhout. The introduction of time-scales in reservoir computing, applied to isolated digits recognition. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, 2007.

[17] Udo Siewert and Welf Wustlich. Echo-state networks with band-pass neurons: Towards generic time-scale-independent reservoir structures. Technical report, Planet GmbH, October 2007.

[18] D. Verstraeten, B. Schrauwen, and D. Stroobandt. Reservoir-based techniques for speech recognition. In *Proceedings of the World Conference on Computational Intelligence*, pages 1050–1053, 2006.

[19] David Verstraeten, Benjamin Schrauwen, Michiel D'Haene, and Dirk Stroobandt. A unifying comparison of reservoir computing methods. *Neural Networks*, 20:391–403, 2007.

[20] O.L. White, D.D. Lee, and H. Sompolinsky. Short-term memory in orthogonal neural networks. *Neural Comput Phys Rev Lett*, 92:148102, 2002.