



## OPEN

# Constructing optimized binary masks for reservoir computing with delay systems

SUBJECT AREAS:

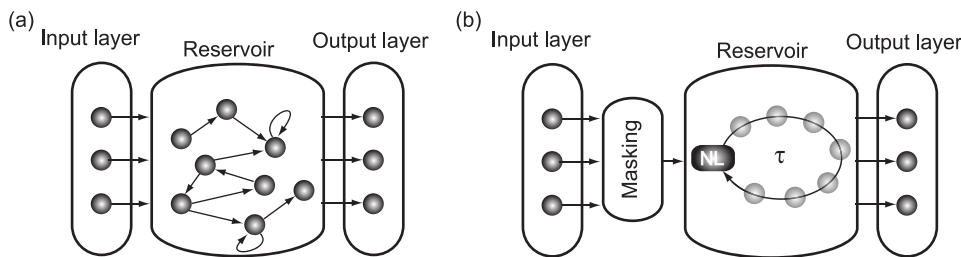
COMPUTATIONAL  
SCIENCE

APPLIED PHYSICS

Lennert Appeltant<sup>1,2</sup>, Guy Van der Sande<sup>1</sup>, Jan Danckaert<sup>1</sup> & Ingo Fischer<sup>2</sup><sup>1</sup>Applied Physics Research Group, Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Elsene, Belgium, <sup>2</sup>Instituto de Física Interdisciplinar y Sistemas Complejos (IFISC-CSIC), Campus Universitat de les Illes Balears, E-07122 Palma de Mallorca, Spain.Received  
22 July 2013Accepted  
11 December 2013Published  
10 January 2014Correspondence and  
requests for materials  
should be addressed to  
L.A. (lappelta@vub.ac.  
be)

**Reservoir computing is a novel bio-inspired computing method, capable of solving complex tasks in a computationally efficient way. It has recently been successfully implemented using delayed feedback systems, allowing to reduce the hardware complexity of brain-inspired computers drastically. In this approach, the pre-processing procedure relies on the definition of a temporal mask which serves as a scaled time-multiplexing of the input. Originally, random masks had been chosen, motivated by the random connectivity in reservoirs. This random generation can sometimes fail. Moreover, for hardware implementations random generation is not ideal due to its complexity and the requirement for trial and error. We outline a procedure to reliably construct an optimal mask pattern in terms of multipurpose performance, derived from the concept of maximum length sequences. Not only does this ensure the creation of the shortest possible mask that leads to maximum variability in the reservoir states for the given reservoir, it also allows for an interpretation of the statistical significance of the provided training samples for the task at hand.**

**R**eservoir Computing is a recently introduced paradigm in the field of machine learning<sup>1-4</sup>. Similar to traditional neural network approaches, an input signal is injected into a network of artificial neurons or nodes, where it is nonlinearly transformed. In this way, the neural network maps the input onto a high-dimensional space, rendering it more suitable for classification. The projection of the input signal onto the high-dimensional state space facilitates classification into different categories drastically compared to using the original inputs. While in traditional neural networks one network does the high-dimensional projection and is being adapted to perform the classification, in reservoir computing the network doing the projection is unaltered and only the readout weights are modified for classification. Consequently, the artificial neural network is split into three separate layers: the input layer, the reservoir layer and the output layer, as illustrated in Fig. 1(a). Exchange of information between and within these layers is defined in the form of weighted connections, resulting in a global network architecture. Each layer handles its specific task in the network. The input layer provides the interface between the real-world signal and the network, while the reservoir layer projects the signal onto a network state better suited for classification. The reasoning behind this is that in a high-dimensional space, constructed by the many nodes in the reservoir, it becomes exponentially more likely that variables with similar features cluster together and can be separated by hyperplanes. Variables that cannot be linearly separated in a low-dimensional space, might be perfectly linearly separable in a high-dimensional space. The interpretation of the high-dimensional mapping and the final classification step is performed by the third layer, the output layer. It is here that the final output series is generated, after a procedure which is called *training*. The training procedure will determine the correct set of weights for the output layer by using examples of the problem to be solved that are fed to the network. A framework has been proposed to define the computational capacity of these systems and to a priori evaluate their suitability for a range of tasks<sup>5</sup>. It was shown that this entire network structure, with sometimes many hundreds or thousands of nodes, can often be replaced by only one single nonlinear node and a delay line<sup>6</sup>, as illustrated in Fig. 1(b). Since the training of the system is performed in the output layer only, it does not alter the dynamics of the reservoir itself. This means that the exact implementation of the reservoir is not constrained to a random network of nodes. Appeltant *et al.* implemented the reservoir layer employing one nonlinear node that performs the transformation and a delayed-feedback line to spatially map temporal information. Along the delay line one can define virtual nodes, playing a role similar to the many nodes present in a traditional network approach. They correspond in fact to a subdivision of the delay line, containing a delayed version of the nonlinear node output. By exploiting the dynamical properties of the system, in particular the different time scales, a virtual interconnection structure can be created, without providing many physical connections between all the virtual nodes. The inertia of the system can introduce coupling between neighbouring



**Figure 1** | Layered representation of a reservoir computing setup. (a) traditional network approach (b) delayed feedback approach with masking procedure.

virtual nodes, while the delayed-feedback line accounts for a strong self-coupling of every node with its older version. When tapping the delay line at the positions of the virtual nodes, the corresponding vector represents the reservoir state. The training procedure is chosen identical to the one used for traditional network approaches of reservoir computing. It was demonstrated that for a number of benchmark tasks the obtained results are comparable or even better than the ones yielded by a large network of randomly interconnected nodes. The fact that not the entire complexity of a random network is needed, but that also simpler architectures suffice to achieve good performance was also shown in<sup>7</sup>. The delayed-feedback concept drastically simplifies the hardware implementation and was later extended to electro-optics<sup>8,9</sup> and even to ultra-fast all-optical implementations<sup>10</sup>. The fact that the input for the virtual nodes needs to be injected in the system via the one nonlinear node, requires, however, pre-processing of the input (time-multiplexing), referred to as masking.

The masking procedure serves two purposes: sequentializing the input and maximizing the effectively used dimensionality of the system. In a traditional network approach all the nodes in the reservoir can be addressed directly via the direct connections from the input layer to the reservoir layer. In the delayed feedback approach the input signal passes through the nonlinear node first, where it undergoes nonlinear transformation and then propagates through the delay line to the virtual nodes. Providing the input signal, with the proper input scaling, to the corresponding virtual node is achieved by time multiplexing the input. Therefore, also the input scaling needs to be imprinted on the input before injection. The result is a piecewise constant input series, with constant intervals corresponding to the separation between the virtual nodes in the delay line.

The delayed coupling in the system provides an infinite-dimensional state space. However, without applying different scaling factors to all the different nodes, the available dimensions are not optimally explored in the reservoir state space. Very similar inputs would result in very similar outputs, which would not span a high-dimensional state space. When using a large number of nodes, the mask containing the scaling factors can often be chosen randomly. For a smaller set of nodes this choice can give bad results and a procedure to reliably assign mask values, such that a maximum diversity in reservoir states is created, is highly desired. Certainly, in some situations one could test different random masks if one only has a single task at hand. Rodan *et al.* already stated that using aperiodic sequences in the input weights deterministically generated from e.g. a chaotic time series could outperform random drawings. Nevertheless, such an approach remains largely heuristic. In this paper, we aim to build a versatile system with good performance regardless of the task or changes in system parameters. The technique presented here, based on insight into the dynamical behaviour of delayed feedback systems, is able to construct the shortest possible mask with excellent performance.

## Results

Here we outline the pre-processing procedure for a scalar input, but the concept can be readily transferred to a vector input with multiple

channels. The goal is to time-multiplex the scaled inputs for the different virtual nodes into one input stream, which then can be injected in the physically present nonlinear node. Any input value  $u(k)$ , originating from a discrete input series or from sampling a continuous input stream, is held constant during the interval  $\tau$ , the delay time of the system. This ensures that the same input value will be fed to the corresponding virtual nodes in the delay line. This piecewise constant function  $I(t)$  with the value  $I(t) = u(k)$  for  $t \in [(k-1)\tau, k\tau]$  is subsequently divided into  $N$  sub-intervals of length  $\theta$ , defining the separation of the virtual nodes. Every interval represents the properly scaled input segment to be fed to one virtual node. For every interval of length  $\theta$ , the separation between the virtual nodes, the input value is multiplied with the scaling factor  $w_i$  of the corresponding virtual node  $i$ , creating the masked input series  $J(t)$  as follows

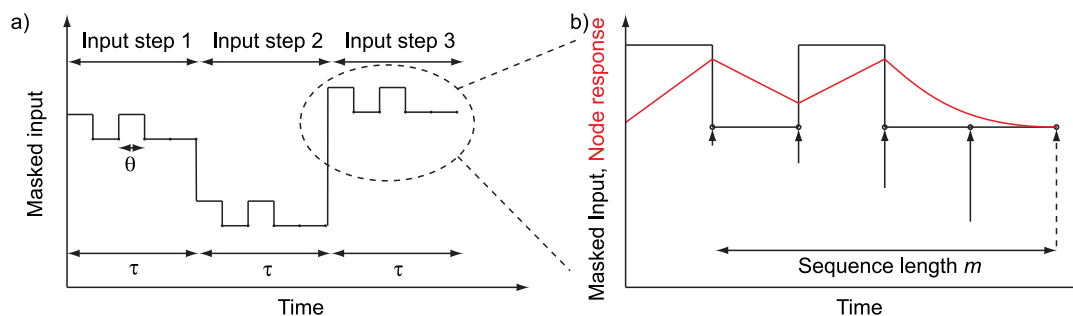
$$J(t_{k,i}) = I(t_{k,i}) \cdot w_i, \quad (1)$$

with  $t_{k,i}$  denoting the time corresponding to input  $k$  and virtual node  $i$ . This can also be expressed as  $t_{k,i} \in [(k-1)\tau + (i-1)\theta, (k-1)\tau + i\theta]$ . The relation between  $\tau$ ,  $\theta$  and  $N$  is given by  $\tau = N \cdot \theta$ .

The importance of the ratio between the time scale  $\theta$  and the characteristic time scale  $T$  of the nonlinear node was pointed out before<sup>6</sup>. Only by choosing  $\theta$  smaller than  $T$ , the node will always be residing in a transient regime, necessary for efficient computational ability. This choice, however, also imposes a constraint on the exact sequence of the scaling factors. The state of a virtual node is determined by the injected input value, by the feedback coming from the delay line, and through inertia by the states of its adjacent neighbouring nodes. If  $\theta$  is chosen to be equal to  $T/\epsilon$ , with  $\epsilon \in \mathbb{N}_0$ , it is reasonable to assume that the state of virtual node  $i$  is determined by the states of virtual nodes  $i-1, i-2, \dots, i-\epsilon, i-m$ , with  $m$  close to  $\epsilon$ . In the following, we refer to  $m$  as the relevant sequence length. Hence, the exact mask sequence is a crucial ingredient in creating diversity in the different reservoir states and fully exploiting the high-dimensionality of the system. This concept is schematically depicted in Fig. 2.

Sufficient diversity can be reached by generating very long mask sequences, containing many possible combinations of the different mask values. However, in terms of efficiency and to perform the training procedure in a not too high-dimensional state space resulting in statistical limitations, it is desirable to construct a mask in the shortest possible way (the smallest possible number of virtual nodes) that contains a maximum of variations in the mask sequences of length  $m$ . When the number of virtual nodes decreases, the length of the delay line also decreases, since they are coupled by the relation  $\tau = N\theta$ . A shorter delay time results in faster information processing.

As an example we consider the situation in which the state of a virtual node is determined by its proper input and the value of only one neighbouring virtual node. To ensure maximum variation in all the obtained virtual node states, all possible sequences of 2 mask values - the virtual node under consideration and its neighbour - need to be present in the mask: 00 01 10 11. However, it becomes clear that the last block, 11, is already present in the combination of block 2 and 3. Hence, the state found for virtual node 5 would be



**Figure 2** | (a) A general scheme of the masked input to be fed to the system is shown. (b) A zoom of the last masked input step is depicted. The state of the present virtual node (dashed arrow) depends in an exponentially decreasing way on the states of previous virtual nodes (solid arrows). The number of relevant nodes to significantly determine a state is called the relevant sequence length  $m$ . The black line represents the masked input and the red line the response of the nonlinear node.

identical to the state of virtual node 8. Instead of explicitly including these 4 blocks in the mask realization, we generate a sequence containing all possible transitions from one mask value to another exactly once: 00110. This mask is shorter and avoids any redundancy in mask value sequences. While for  $m = 2$  this is straightforward, it becomes more involved to construct these sequences for larger  $m$ .

In order to ensure that the dimensionality of the system is always fully explored in the shortest possible way, we outline a method that guarantees an optimal choice for the mask values. We conjecture that the series of mask values should contain all possible patterns, given a certain sequence length  $m$  of the pattern. If (following the example of Fig. 2) the node state is strongly dependent on the states that were obtained for the 4 previous adjacent virtual nodes, we believe no more information can be extracted than given by all possible length-4 sequences of the two mask values. Moreover, in order to construct the most efficient mask, we assume all of them should occur exactly once. This can be done by using a modification of what is called maximum length sequences. In maximum length sequences, a series of values is generated that contains all but one possible bit patterns of an  $m$ -bit block in a ring structure. Only the all-zero sequence is not included. For detailed descriptions on theory around and construction of these sequences we refer to literature<sup>11–13</sup>. Here, only binary masks are considered, but the approach might be extended to multi-level masks using  $m$ -ary maximum length sequences<sup>14</sup>.

Since, in the case of the constructed masks, all bit patterns need to be present for one input step of length  $\tau$ , the ring structure is not applicable here. In general, when all possible realizations of  $m$  bits are required in the mask, the minimal mask length is exactly  $2^m + m - 1$ . This extra length of  $m$  bits, as compared to the length of  $2^m - 1$  for a pure maximum length sequence, originates from adding one 0 (the combination with  $m$  zeros is not present in a maximum length sequence) and from adding the last  $m - 1$  bits of the sequence to the beginning of the series (because the mask is not a ring structure for one input step).

The key idea of our procedure is not to do a numerical optimization of the reservoir for a certain task, but to realize an optimality in terms of diversity of  $m$ -bit sequences, therefore obtaining a versatile, not task-specific, high-dimensional projection. An efficient training algorithm requires every feature to be present in the training data only once. Given a relevant sequence length  $m$  (fully determined by the choice of the time scale  $\theta$ ) we guarantee that all state-space directions are maximally explored. Every possible sequence is present in the constructed mask, leading to a maximum variety in the reservoir states. By employing maximum length sequences, this concept is implemented in a way that is completely independent of the chosen nonlinearity or benchmark. Still, the best-performing random masks will be able to match the optimal masks in terms of performance. However, our technique is not only oriented towards

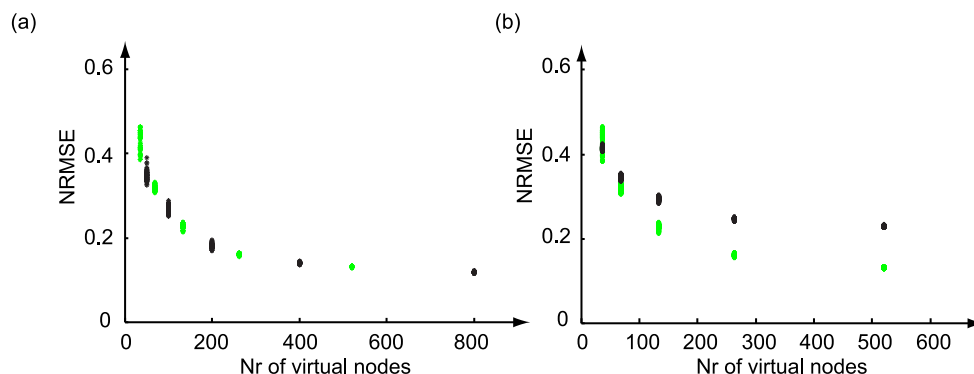
increasing performance, its goal is also to eliminate the standard deviation on performance. For every result generated with a random mask, one needs to perform several runs of the experiment to have an indication whether the chosen mask is a good or a bad performer. We have eliminated this trial and error, which is also detrimental for hardware implementations, and we guarantee that the constructed mask is among the best performing ones, even with one single run. The difference in standard deviation becomes even of more important when going to shorter mask sequences. In delayed-feedback reservoir computing the time needed to process a signal is related to the length of the mask, hence shorter masks result in faster information processing. Also, constructing the mask such that it optimally explores the full dimensionality of the system, ensures good performance for any task or any parameter set, allowing for versatility.

Here, a delayed-feedback Mackey-Glass oscillator<sup>15</sup> is used, originally introduced as a model of blood cell regulation, but other nonlinear nodes are expected to perform similarly. The equation of the system for this nonlinearity type is

$$\dot{x}(t) = -x(t) + \eta \frac{x(t-\tau) + \gamma J(t)}{1 + [x(t-\tau) + \gamma J(t)]^p}, \quad (2)$$

with  $\dot{x}$  representing the derivative of  $x$  with respect to time,  $\eta$  the feedback strength,  $\gamma$  the input scaling for masked input  $J$ , and  $\tau$  the delay time. Eq. (2) is numerically integrated using Heun's method with a time step of 0.1. By replacing the values 0 and 1 of the previous section with the low and high value of the binary mask, respectively, we can construct a mask with all possible mask value sequences of length  $m$  present. For both tasks the reservoir responses to 4 input samples of length 4000 each were collected and the training was performed using a Moore-Penrose pseudo inverse and 2-fold cross-validation. Overfitting is avoided using Tikhonov regularization and all the reported errors are test errors.

To validate the strategy, two benchmark tasks are solved. Using these optimal masks, a performance plot is shown for the Mackey-Glass nonlinearity in Fig. 3(a), where the number of virtual nodes is varied and the performance on the NARMA10 task is quantified via the NRMSE. The NARMA10 task is a nonlinear system modelling task which was originally introduced in<sup>16</sup> and which is commonly used in the field of reservoir computing. The parameters of the Mackey Glass system are chosen close to the optimal settings for this task:  $\eta = 0.5$ ,  $\gamma = 0.001$ ,  $p = 1$  and  $\theta = 0.2$ , while the two mask values are  $\pm 1$ . The value of  $\tau$  is determined by the relation  $\tau = N\theta$ , with  $N$  being the number of virtual nodes. For all experiments these parameter settings correspond to a zero steady-state solution in absence of the input. The green dots represent the scoring of 100 constructed optimal masks, while the black dots mark the scoring of 100 randomly chosen binary masks. The random masks could include the



**Figure 3 | Performance plot NARMA10 for a Mackey-Glass nonlinearity type with parameter settings:  $\eta = 0.5$ ,  $\gamma = 0.001$  and  $p = 1$ .** (a) Random and constructed optimal masks for  $\theta = 0.2$ : the black points denote the scoring of the random masks, while the green points indicate the NRMSE obtained for constructed optimal masks. (b) Constructed optimal masks for different  $\theta$ : the green points denote the scoring of the constructed optimal masks for  $\theta = 0.2$ , while the black points indicate the NRMSE obtained for the same optimal masks for  $\theta = 1$ . In both plots 100 masks were generated for every node number.

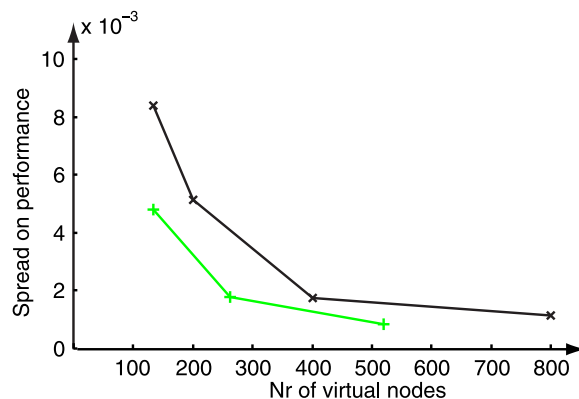
constructed mask sequence, as well as the sequence with all mask values being identical, corresponding to the case of no masking. The latter realization scores worse than a purely linear reservoir. This illustrates that the total standard deviation for the random masks can be very large and is much larger than shown here. First, it can be noted that for the obtained NRMSE a saturation can be observed when increasing the number of virtual nodes up to 263 (8-bit sequences) or more. Second, a higher virtual node number results in a smaller standard deviation on the performance. This complies with the fact that the virtual node separation is chosen to be 0.2, corresponding to 5 virtual nodes per characteristic time scale of the nonlinear node. The response time of the node corresponds to 5 virtual nodes, hence it is reasonable to assume that combinations with a total length of a little more than the response time are significant to determine the state of a virtual node. Any older state has no significant influence on the present node state and cannot create more variation in the reservoir states. For smaller virtual node numbers, not all possible bit combinations of 8 or more bits are present. For some of the randomly chosen masks there are more relevant sequences lacking than for others, resulting in a larger standard deviation on the error. When increasing the number of nodes, all standard deviation on the performance disappears since all possible mask value patterns are included and even a certain redundancy is introduced. We remark that the average performance is very comparable for random and constructed masks, but in the case of the constructed masks the obtained performance is more consistent. The results are not only of excellent quality, but they are also reliable, eliminating the need for trial and error in the selection of the mask, as done otherwise.

In Fig. 3(b) the performance of the optimal masks is shown for two values of  $\theta$ , the virtual node separation. The green points denote the score of the system for  $\theta = 0.2$  and the black points for  $\theta = 1$ . Here, the importance of the sequence length is clearly illustrated. The standard deviation on the values obtained for  $\theta = 1$  is significantly lower, because adjacent virtual node contribute less to the creation of a virtual node state. Only the very nearest neighbours play a role. When all possible patterns of longer sequence lengths are included as well, no real improvement can be achieved in terms of standard deviation. The relevant combinations of mask values are already present. The choice of  $\theta$  determines the connectivity in the system, leading to a certain error that can be achieved (for this task  $\theta = 0.2$  is optimal). This explains why the optimal NRMSE is higher for all sequences for  $\theta = 1$ . Once the desired virtual node separation is selected, our procedure constructs an optimal mask, such that the system's high-dimensionality is fully explored and a minimum value of  $\tau$  can be determined, which yields a maximum variability in the

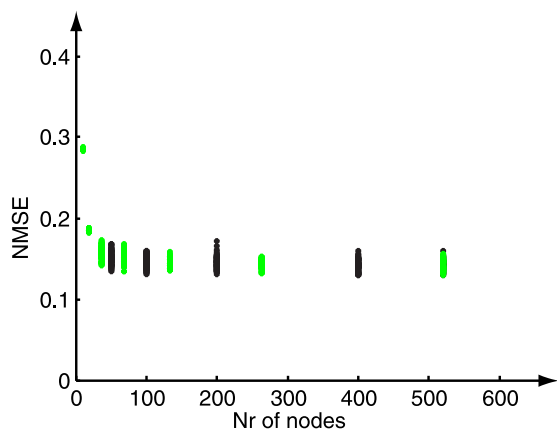
virtual node states. Choosing  $\theta$  and the sequence length  $m$  go hand in hand. Therefore, the effort for optimising the reservoir can now be greatly reduced.

In Fig. 4 only the standard deviation on performance is shown, for a situation similar to that one of Fig. 3(a). The black crosses, indicating the standard deviation on performance found for random masks, are systematically positioned at higher values than the green pluses, corresponding to the standard deviation found for optimal masks, constructed using maximum length sequences. We remark that the standard deviation shown for the random masks is based on the relatively wellperforming masks of Fig. 3. They do not include possible random realizations with alternating or all-identical mask patterns, which would significantly increase the standard deviation.

When the same approach is applied to the Santa Fe Laser time series prediction task, a similar performance on virtual node number is found, as shown in Fig. 5. Again, the number of virtual nodes is varied, but this time the performance is shown as an NMSE. The black data points denote the error of the randomly chosen masks and the green data points represent the corresponding errors for the optimal masks. Here, saturation of the performance already occurs for 36 virtual nodes. It is observed that for this benchmark the standard deviation is slightly larger and more importantly, a mask consisting of only short patterns is sufficient. The standard deviation



**Figure 4 | Plot of the standard deviation on performance for NARMA10 with a Mackey-Glass nonlinearity type with parameter settings:  $\eta = 0.5$ ,  $\gamma = 0.001$ ,  $p = 1$  and  $\theta = 0.2$ .** The black crosses denote the standard deviation on performance of the random masks, while the green pluses indicate the obtained standard deviation for constructed optimal masks. For every virtual node number 100 masks were generated.



**Figure 5 | Performance plot for the Santa Fe laser prediction task for random and optimal masks.** A Mackey-Glass nonlinearity type is used, with parameter settings:  $\eta = 0.5$ ,  $\gamma = 0.001$ ,  $p = 1$  and  $\theta = 0.2$ . The black points denote the scoring of the random masks, while the green points indicate the NMSE obtained for optimal masks. For every scanned node number 100 masks were generated.

hardly decreases when working with higher node numbers, on the contrary, the standard deviation increases again for 520 nodes. This can be explained by the fact that for this task two different effects are observed. The standard deviation decreases with the number of virtual nodes because of variability in the reservoir states, but for high node numbers the standard deviation increases due to the statistical insufficiency of the training procedure. When scaling up the dimensionality, i.e. the number of virtual nodes used for training, the training samples to be considered should scale up in length as well. While for the NARMA10 task the training samples can be chosen arbitrarily long, this is not the case for the Santa Fe task. The number of samples is fixed and, as can be deduced from the results in Fig. 5, not sufficient for 520 nodes or more. In case a random mask would have been used, no distinction could have been made between the standard deviation due to the quality of the randomly drawn values of the mask and the standard deviation due to training statistics. Through optimally constructing the mask based on the sequence length one can conclusively estimate the limits of the training data by evaluating the remaining spread. Only for the optimized masks it would disappear for a sufficient  $m$  and a sufficient set of training data.

## Discussion

We have introduced a method to pre-process the input of a delayed-feedback reservoir such that the high-dimensionality of the system is optimally explored in the sense that a maximum variability is created in the reservoir states, while reducing the number of virtual nodes to a minimum. Next to faster signal processing, another advantage is the elimination of the standard deviation on the performance due to the specific mask realization. This implies that there is no need for trial and error during selection of the mask, which is particularly important when realizing full hardware implementations of the concept. The only resulting variance in performance is due to the training samples and the training algorithm itself. Hence, this method allows us in addition to detect training samples that are not statist-

ically significant for solving the task at hand, a conclusion that can never be drawn with certainty for random masks.

1. Jaeger, H. & Haas, H. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science* **304**, 78–80 (2004).
2. Maass, W., Natschläger, T. & Markram, H. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.* **14**, 2531–2560 (2002).
3. Steil, J. J. Backpropagation-decorrelation: Online recurrent learning with  $O(N)$  complexity. vol. 2, 843–848, IEEE doi:10.1109/IJCNN.2004.1380039 (Paper presented at 2004 IEEE IJCNN, Budapest, 2004 July 25–29).
4. Verstraeten, D., Schrauwen, B., D'Haene, M. & Stroobandt, D. An experimental unification of reservoir computing methods. *Neural Networks* **20**, 391–403 (2007).
5. Dambre, J., Verstraeten, D., Schrauwen, B. & Massar, S. Information processing capacity of dynamical systems. *Sci. Rep.* **2**, 1–7 (2012).
6. Appeltant, L. *et al.* Information processing using a single dynamical node as complex system. *Nat. Commun.* **2**, 468 (2011).
7. Rodan, A. & Tino, P. Minimum complexity echo state network. *IEEE T. Neural Networ.* **22**, 131–144 (2011).
8. Larger, L. *et al.* Photonic information processing beyond turing: an optoelectronic implementation of reservoir computing. *Opt. Express* **5**, 188–200 (2012).
9. Paquot, Y. *et al.* Optoelectronic reservoir computing. *Sci. Rep.* **2**, 287 (2012).
10. Brunner, D., Soriano, M. C., Mirasso, C. R. & Fischer, I. Parallel photonic information processing at gigabyte per second data rates using transient states. *Nat. Commun.* **4**, 1364 (2013).
11. Debruijn, N. A combinatorial problem. *P. K. Ned. Akad. Wetensc.* **49**, 758–764 (1946).
12. Debruijn, N. Acknowledgement of priority to c. flye saintemarie on the counting of circular arrangements of  $2n$  zeros and ones that show each  $n$ -letter word exactly once. *T.H.-Report 75-WSK-06, Technological University Eindhoven* (1975). URL alexandria.tue.nl/repository/books/252901.pdf, Date Accessed Nov. 18, 2013.
13. Davies, W. D. T. Generation and properties of maximum-length sequences. *Control* **10**, 302–304; 364–365; 431–433 (1966).
14. Lablans, P. Us patent: Ternary and multi-value digital signal scramblers, decramblers and sequence generators (2011). URL <http://www.faq.s.org/patents/app/20110170697>, Date Accessed Nov. 18, 2013.
15. Mackey, M. C. & Glass, L. Oscillation and chaos in physiological control systems. *Science* **197**, 287–289 (1977).
16. Atiya, A. F. & Parlos, A. G. New results on recurrent network training: unifying the algorithms and accelerating convergence. *IEEE T. Neural Netw.* **11**, 697–709 (2000).

## Acknowledgments

We thank the members of the PHOCUS consortium for helpful discussions. Stimulating discussions are acknowledged with Gordon Pipa. This research was partially supported by the Belgian Science Policy Office, under grant IAP P7/35 Photonics@be, by FWO(Belgium), MICINN (Spain), Comunitat Autònoma de les Illes Balears, FEDER, and the European Commission under Projects TEC2012-36335 (TRIPHOP), Grups Competitius and EC FP7 Projects PHOCUS (Grant No. 240763).

## Author contributions

L.A. and I.F. developed the concept, L.A. and G.V.d.S. performed the numerical simulations. L.A., G.V.d.S., J.D. and I.F. discussed and interpreted the results. L.A., G.V.d.S. and I.F. wrote the main manuscript text.

## Additional information

**Competing financial interests:** The authors declare no competing financial interests.

**How to cite this article:** Appeltant, L., Van der Sande, G., Danckaert, J. & Fischer, I. Constructing optimized binary masks for reservoir computing with delay systems. *Sci. Rep.* **4**, 3629; DOI:10.1038/srep03629 (2014).



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported license. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0>