

Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless telecommunication

Herbert Jaeger*, Harald Haas, International University Bremen, Bremen, Germany.

*to whom correspondence should be addressed, email: h.jaeger@iu-bremen.de

Abstract. We present a method for learning nonlinear systems, echo state networks (ESNs). ESNs employ artificial recurrent neural networks in a way that has recently been proposed independently as a learning mechanism in biological brains. The learning method is computationally efficient and easy to use. On a benchmark task of predicting a chaotic time series, accuracy is improved by a factor of 2400 over previous techniques. The potential for engineering applications is illustrated by equalizing a communication channel, where the signal error rate is improved by two orders of magnitude.

Nonlinear dynamical systems abound in the sciences and in engineering. If one wishes to simulate, predict, filter, classify or control such a system, one needs an executable system model. However, it is often infeasible to obtain analytical models. In such cases one has to resort to blackbox models, which ignore the internal physical mechanisms and instead only reproduce the outwardly observable input-output behavior of the target system.

If the target system is linear, efficient methods for blackbox modeling are available. Most technical systems however become nonlinear if operated at higher operational points (that is, closer to saturation). Although this might lead to cheaper and more energy-efficient designs, it is not done because the resulting nonlinearities cannot be harnessed. Many biomechanical systems use their full dynamic range (up to saturation) and thereby become lightweight and energy-efficient – and thoroughly nonlinear.

Here we present an approach to learning blackbox models of nonlinear systems, echo state networks (ESNs). An ESN is an artificial recurrent neural network (RNN). RNNs are characterized by feedback ("recurrent") loops in their synaptic connection pathways. They can maintain an ongoing activation even in the absence of input and thus exhibit dynamic memory. Biological neural networks are typically recurrent. Like biological neural networks, an artificial RNN can learn to mimic a target system – in principle with arbitrary accuracy (*1*). Several learning algorithms are known (*2–4*) that incrementally adapt the synaptic weights of an RNN in order to tune it toward the target system. These algorithms have not been widely employed in technical applications because of slow convergence and suboptimal solutions (*5,6*). The ESN approach differs from these methods in that a large RNN is used (order of 50 to 1000 neurons, previous techniques typically use 5 to 30 neurons) and in that only the synaptic connections from the RNN to the output readout neurons are modified by learning (previous techniques tune all synaptic connections, Fig. 1). Because there are no cyclic dependencies between the trained readout connections, training an ESN becomes a simple linear regression task.

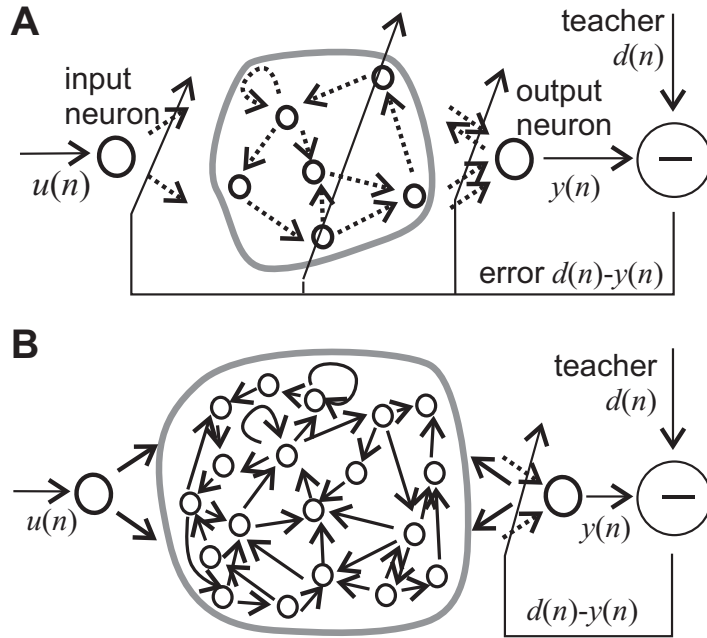


Fig. 1. (A) Schema of previous approaches to RNN learning. **(B)** Schema of ESN approach. Solid bold arrows: fixed synaptic connections, dotted arrows: adjustable connections. Both approaches aim at minimizing the error $d(n) - y(n)$, where $y(n)$ is the network output and $d(n)$ is the "teacher" time series observed from the target system.

We illustrate the ESN approach on a task of chaotic time series prediction (7). The Mackey-Glass system (8) is a standard benchmark system for time series prediction studies. It generates a subtly irregular time series (Fig. 1A). The prediction task has two steps: (i) use an initial teacher sequence generated by the original MGS to learn a blackbox model M of the generating system, (ii) use M to predict the value of the sequence some steps ahead.

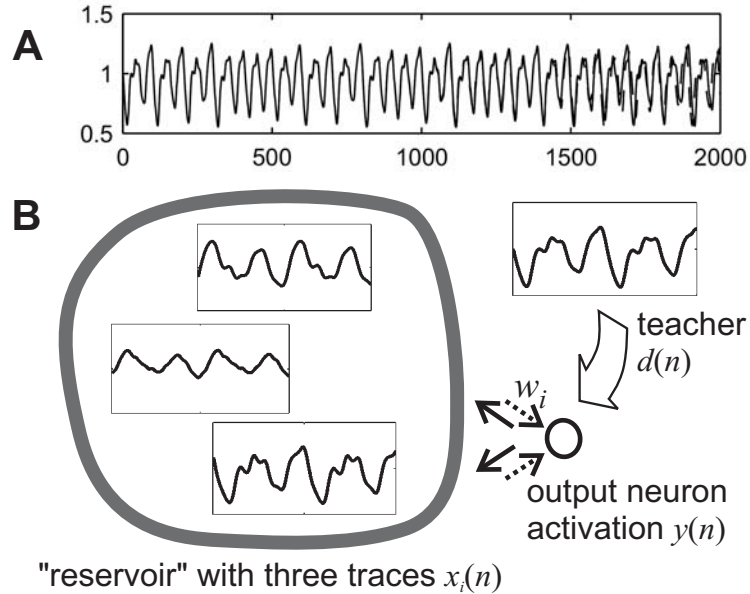


Fig. 2. (A) Prediction output of the trained ESN (dotted) overlaid with the correct continuation (solid). (B) Learning the MG attractor. Three sample activation traces of internal neurons are shown. They "echo" the teacher signal $d(n)$. After training, the desired output is re-created from the echo signals through output connections (dotted arrows) whose weights w_i are the result of the training procedure.

First we created a random RNN with 1000 neurons (called the "reservoir") and one output neuron. Importantly, the output neuron was equipped with random connections that project back into the reservoir (Fig. 1B). A 3000 step teacher sequence $d(1), \dots, d(3000)$ was generated from the MGS equation and fed into the output neuron. This excited the internal neurons through the output feedback connections. After an initial transient, they started to exhibit systematic individual variations of the teacher sequence (Fig. 1B).

The fact that the internal neurons display systematic variants of the exciting external signal is constitutional for ESNs: the internal neurons must work as "echo functions" for the driving signal. Not every randomly generated RNN has this property, but it can effectively be built into a reservoir (supporting online text).

It is important that the "echo" signals be richly varied. This was ensured by a sparse interconnectivity of 1% within the reservoir: this condition lets the reservoir decompose into many loosely coupled subsystems, establishing a richly structured reservoir of excitable dynamics.

After time $n = 3000$, output connection weights w_i ($i = 1, \dots, 1000$) were computed (dashed arrows in Fig. 1B) from the last 2000 steps $n = 1001, \dots, 3000$ of the training run such that the training error $\text{MSE}_{\text{train}} = 1/2000 \sum_{n=1001}^{3000} \left(d(n) - \sum_{i=1}^{1000} w_i x_i(n) \right)^2$ was minimized [$x_i(n)$: activation of the i -th internal neuron at time n]. This is a simple linear regression.

With the new w_i in place, the ESN was disconnected from the teacher after step 3000 and left running freely. A bidirectional dynamical interplay of the network-generated output signal with the internal signals $x_i(n)$ unfolded. The output signal $y(n)$ was created from the internal neuron activation signals $x_i(n)$ through the trained connections w_i , by $y(n) = \sum_{i=1}^{1000} w_i x_i(n)$. Conversely, the internal signals were "echoed" from that output signal through the fixed output feedback connections (supporting online text).

For testing, a 84-step continuation $d(3001), \dots, d(3084)$ of the original signal was computed for reference. The network output $y(3084)$ was compared with the correct continuation $d(3084)$. Averaged over 100 independent trials, an normalized root mean square error

$$\text{NRMSE} = \left(\sum_{j=1}^{100} (d_j(3084) - y_j(3084))^2 / 100 \sigma^2 \right)^{1/2} \approx 10^{-4.2}$$

was obtained [d_j and y_j teacher

and network output in trial j , σ^2 variance of MGS signal], improving the best previous techniques (9–15), which used training sequences of length 500 to 10000, by a factor of 700. If the prediction run was continued, deviations typically became visible after about 1300 steps (Fig. 1A). With a refined variant of the learning method (7) the improvement factor rises to 2400. Models of similar accuracy were also obtained for other chaotic systems (supporting online text).

The main reason for the jump in modelling accuracy is that ESNs capitalize on a massive short-term memory. We showed analytically (16) that under certain conditions an ESN of size N may be able to "remember" in the order of the last N inputs. This information is more massive than the information used in other techniques (supporting online text).

We now illustrate the approach in a task of practical relevance, namely, the equalization of a wireless communication channel (7). The essentials of equalization are as follows. A sender wants to communicate a symbol sequence $s(n)$. This sequence is first transformed into an analog envelope signal $d(n)$, then modulated on a high-frequency carrier signal and transmitted, then received and demodulated into an analog signal $u(n)$ which is a corrupted version of $d(n)$. Major sources of corruption are (i) noise (thermal or due to interfering signals), (ii) multipath propagation, which leads to a superposition of adjacent symbols (intersymbol interference), and nonlinear distortion induced by operating the sender's power amplifier in the high gain region. In order to avoid the latter, the actual power amplification is run well below the maximum amplification possible, thereby incurring a significant loss in energy efficiency which is clearly undesirable in cellphone and satellite communications. The corrupted signal $u(n)$ is then passed through an equalizing filter whose output $y(n)$ should restore $u(n)$ as closely as possible to $d(n)$. Finally, the equalized signal $y(n)$ is converted back into a symbol sequence. The quality measure for the entire process is the fraction of incorrect symbols finally obtained (symbol error rate).

In order to compare the performance of an ESN equalizer with standard techniques, we took a channel model for a nonlinear wireless transmission system from a study (17) that compared three customary nonlinear equalization methods, namely a linear decision feedback equalizer (DFE), which is actually a nonlinear method; a Volterra DFE; and a bilinear DFE. The model equation featured intersymbol interference across 10 consecutive symbols, a second-order and a third-order nonlinear distortion, and additive white Gaussian noise. All methods investigated in that study had 47 adjustable parameters and used sequences of 5000 symbols for training. In order to make the ESN equalizer comparable with the equalizers studied in (17), we took ESNs with a reservoir of 46 neurons (which is small for the ESN approach), which yielded 47

adjustable parameters (the 47th comes from a direct connection from the input to the output neuron).

We carried out numerous learning trials (7) to obtain ESN equalizers, using an online learning method (a version of the recursive least square algorithm known from linear adaptive filters) to train the output weights on 5000 step training sequences. We chose an online adaptation scheme (17) here because the methods in (17) were online adaptive, too, and because wireless communication channels mostly are time-varying, such that an equalizer must adapt to changing system characteristics. The entire learning-testing procedure was repeated for signal-to-noise ratios ranging from 12 to 32 db. Fig. 3 compares the average symbol error rates obtained to the results reported in (17), showing an improvement of two magnitudes for high signal-to-noise ratios.

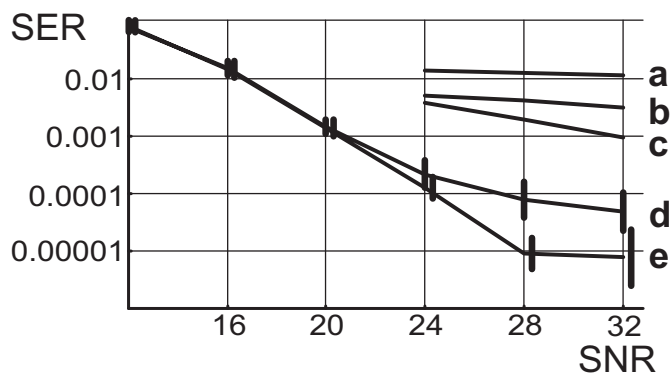


Fig. 3. Using an ESN for nonlinear channel equalization: results. Plot shows signal error rate SER vs. signal-to-noise ratio SNR. **a.** linear DFE, **b.** Volterra DFE, **c.** bilinear DFE (**a – c** taken from (18)). **d.** Blue line represents average ESN performance with randomly generated "reservoirs" (error bars: variation across networks). **e.** Green line indicates performance of best network chosen from the networks averaged in **d** (error bars: variation across learning trials).

For tasks with multi-channel input and/or output, the ESN approach can be accommodated simply by adding more input or output neurons (19,20).

ESNs can be taken to all basic tasks of signal processing and control, including time series prediction, inverse modeling, pattern generation, event detection and classification, modeling distributions of stochastic processes, filtering and nonlinear control (16,18,19,20). Because a single learning run takes only a few seconds (or minutes for very large datasets and networks), engineers can test out variants at a high turnover rate, a crucial factor for practical usability.

ESNs have been developed from a mathematical and engineering perspective, but exhibit typical features of biological RNNs: a large number of neurons, recurrent pathways, sparse random connectivity and local modification of synaptic weights. The idea of using randomly connected RNNs to represent and memorize dynamic input in network states has frequently been explored in specific contexts, for instance artificial intelligence models of associative memory (21),

models of prefrontal cortex function in sensory-motor sequencing tasks (22), models of birdsong (23), models of the cerebellum (24) or general computational models of neural oscillators (25). Many different learning mechanisms were considered, mostly within the RNN itself. The contribution of the ESN is to have elucidated the mathematical properties of large RNNs such that they can be used with a linear, trainable readout mechanism for general blackbox modeling. An approach essentially equivalent to ESNs, liquid state networks (26,27), has been developed independently to model computations in cortical microcircuits. Recent findings in neurophysiology suggest that the basic ESN/liquid state network principle seems not uncommon in biological networks (28-30) and could eventually be exploited to control prosthetic devices by signals collected from a collective of neurons (31).

References and notes

1. K.-I. Funahashi, Y. Nakamura, *Neural Networks* **6**, 801 (1993).
2. D. Zipser, R. J. Williams, *Neural Comput.* **1**, 270 (1989).
3. P. J. Werbos, *Proc. IEEE* **78**(10), 1550 (1990).
4. L.A. Feldkamp, D. V. Prokhorov, C. F. Eagen, F. Yuan, in *Nonlinear Modeling: Advanced Black-Box Techniques*, J. A. K. Suykens, J. Vandewalle, J., Eds. (Kluwer 1998) pp. 29-54.
5. K. Doya, in *The Handbook of Brain Theory and Neural Networks*, M.A. Arbib, Ed. (MIT Press 1995) pp. 796-800.
6. H. Jaeger, "Tutorial on training recurrent neural networks" (GMD-Report 159, German National Research Institute for Computer Science, 2002).
ftp://borneo.gmd.de/pub/indy/publications_herbert/CompleteTutorialTechrep.pdf
7. Methods are available as supporting material on *Science Online*.
8. M.C. Mackey, L. Glass, *Science* **197**, 287(1977).
9. J. Vesanto, in *Proc. WSOM'97* (1997).
<http://www.cis.hut.fi/projects/monitor/publications/papers/wsom97.ps.zip>
10. L. Chudy, I. Farkas, *Neural Network World* **8**, 481 (1998).
11. H. Bersini, M. Birattari, G. Bontempi, in *Proc. IEEE World Congr. on Computational Intelligence (IJCNN '98)* pp. 2102-2106 (1997).
ftp://iridia.ulb.ac.be/pub/lazy/papers/IridiaTr1997-13_2.ps.gz
12. T. M. Martinetz, S. G. Berkovich, K. J. Schulten, *IEEE Trans. Neural Networks* **4**, 558 (1993).
13. Yao, X. & Liu. Y. *IEEE Trans. Neural Networks* **8**, 694 (1997).
14. F. Gers, D. Eck, J. F. Schmidhuber, "Applying LSTM to time series predictable through time-window approaches" (IDSIA-IDSIA-22-00, 2000).
<http://www.idsia.ch/~felix/Publications.html>
15. J. McNames, J. A. K. Suykens, J. Vandewalle, *Int. J. Bifurcation Chaos* **9**, 1485 (1999).

16. H. Jaeger, "The echo state approach to analysing and training recurrent neural networks" (GMD-Report 148, German National Research Institute for Computer Science 2001).
ftp://borneo.gmd.de/pub/indy/publications_herbert/EchoStatesTechRep.pdf
17. V. J. Mathews, J. Lee, in *Advanced Signal Processing: Algorithms, Architectures, and Implementations V (Proc. SPIE Vol. 2296)*, (SPIE, San Diego, CA, 1994), pp. 317-327.
18. H. Jaeger, in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, K. Obermayer, Eds. (MIT Press, Cambridge, MA, 2003) pp. 593-600
19. H. Jaeger, "Short term memory in echo state networks" (GMD-Report 152, German National Research Institute for Computer Science 2002).
ftp://borneo.gmd.de/pub/indy/publications_herbert/STMEchoStatesTechRep.pdf
20. J. Hertzberg, H. Jaeger, F. Schönherr, in *Proc. 15th Europ. Conf. on Art. Int. (ECAI 02)*, F. van Harmelen, Ed., (IOS Press, Amsterdam, 2002), pp. 708-712.
<http://www.ais.fhg.de/~schoenhe/papers/ECAI02.pdf>
21. G. E. Hinton, in *Parallel Models of Associative Memory*, G. E. Hinton, J. A. Anderson, Eds. (Erlbaum, Hillsdale, NJ, 1981) pp.161-187.
22. D. G. Beiser, J. C. Houk, *J Neurophysiol* **79**: 3168 (1998).
23. S. Dehaene, J.-P. Changeux, J.-P. Nadal, *Proc. Natl. Acad. Sci. USA* **84**, 2727 (1987).
24. M. Kawato, in *The Handbook of Brain Theory and Neural Networks*, M. Arbib, Ed. (MIT Press, 1995), pp. 172-178.
25. K. Doya, S. Yoshizawa, *Neural Networks* **2**, 375 (1989).
26. W. Maass, T. Natschläger, H. Markram, *Neural Computat.* **14**, 2531 (2002).
27. W. Maass, T. Natschläger, H. Markram, in *Computational Neuroscience: A Comprehensive Approach*, J. Feng, Ed. (Chapman & Hall/CRC, 2003), 575-605. [Article](#)
28. G. B. Stanley, F. F. Li, Y. Dan, *J. Neurosci.* **19**, 8036 (1999). [Article](#)
29. G. B. Stanley, *Neurocomputing* **38-40**, 1703 (2001).
30. W. M. Kistler, Ch. I. de Zeeuw, *Neural Comput.* **14**, 2597 (2002). [Article](#)
31. S. Mussa-Ivaldi, *Nature* **408** (6810) 361 (2000).
32. The first author thanks Thomas Christaller for unfaltering support and Wolfgang Maass for friendly co-operation. International patents are claimed by Fraunhofer AIS (PCT/EP01/11490).