Improving performance of opto-electronic reservoir computers with online learning

P. Antonik,[1] F. Duport,[2] A. Smerieri,[2] M. Hermans,[1] M. Haelterman,[2] S. Massar,[1]

[1] Laboratoire d'Information Quantique (LIQ), Université libre de Bruxelles (U.L.B.), 50 Avenue F. D. Roosevelt, CP 225, B-1050 Bruxelles, Belgium
[2] Service OPERA-Photonique, Université libre de Bruxelles (U.L.B.), 50 Avenue F. D. Roosevelt, CP 194/5, B-1050 Bruxelles, Belgium

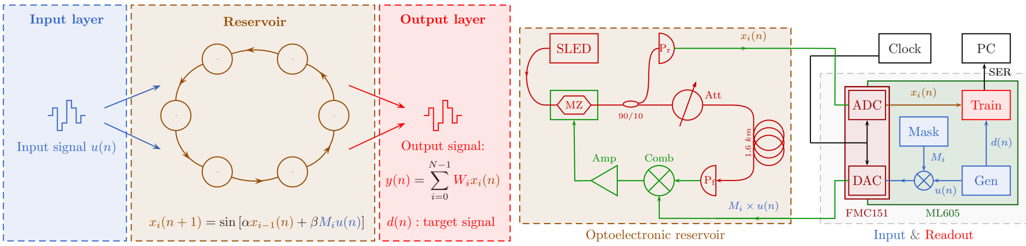
Reservoir Computing is a bio-inspired computing paradigm for processing time dependent signals. Its performance on a series of benchmark tasks is equal and sometimes better than other digital algorithms. It is particularly well suited for analog implementations, but these are often constrained by the offline training algorithms commonly employed. Here we present an opto-electronic reservoir computer trained online by an FPGA chip. We test the system on a channel equalisation task and report error rates up to two orders of magnitude lower than previous implementations. Moreover, we demonstrate how online training allows the computer to adapt to a non-stationary task.

Introduction

Reservoir Computing (RC) is a set of methods for designing and training artificial recurrent neural networks [1]. Although it greatly simplifies the training process, the resulting performances are equal to other algorithms on a series of benchmark tasks [2]. This paradigm is well suited for analog implementations: several opto-electronic [3–5] and all-optical reservoir computers [6–8] have been reported.
The performance of a reservoir computer greatly relies on the training technique. Up to now experimental reservoir computers have been trained using offline learning methods [3, 4, 6–10]. An alternative approach is to use online training in which the readout weights are progressively adapted in real time [11]. We have investigated an example of such algorithms in simulations [12] and experimentally [13], with performance close to previous offline-trained implementations. These works are based on the same opto-electronic reservoir as in [3], but trained online by an FPGA board.
In this work we present important improvements over [12, 13]. Further experimental investigations allow us to significantly increase the performance on the nonlinear channel equalisation task [14]. We also consider a simple case of a variable communication channel and show that online training can deal with non-stationary problems, whereas it would be difficult to adapt offline training methods to such tasks.

Experimental Reservoir Computing

A general reservoir computer is described in [2]. In our implementation, depicted in figure 1(a), we use a sine function $f = \sin(x)$, as in [3, 4], and a ring topology [15] to simplify the interconnection matrix, so that only the first neighbour nodes are connected.



(a) (b)

Figure 1: **(a)** Schematic representation of our reservoir computer. It contains $N = 50$ internal variables $x_i(n)$ evolving in discrete time $n \in \mathbb{Z}$, $\sin(x)$ is the nonlinear function, $u(n)$ is an external signal that is injected into the system, α and β parameters are used to adjust the feedback and the input signals, respectively, and M_i is the input mask, drawn from a uniform distribution over the the interval $[-1, +1]$. The readout weights w_i are trained online by minimising the error between the output signal $y(n)$ and the target signal $d(n)$. **(b)** Schematic representation of the experimental setup.

The reservoir computer is trained online using the simple gradient descent algorithm [16], with the following rule for updating the readout weights:

$$w_i(n+1) = w_i(n) + \lambda(n)\left(d(n) - y(n)\right)x_i(n), \tag{1}$$

where $\lambda(n)$ is the step size, used to control the learning rate. It starts at $\lambda(0) = 0.4$, and then gradually decreases to zero following $\lambda(n+1) = \gamma\lambda(n)$, with $\gamma < 1$.
Figure 1(b) depicts our experimental setup containing the opto-electronic reservoir and the FPGA board. A personal computer is used to control the experiment, scan the parameters α, β and γ and collect the results.
The opto-electronic reservoir is a replica of [3]. The reservoir states are encoded into the intensity of the incoherent light signal, produced by a superluminiscent diode (SLED). The Mach Zehnder intensity modulator (MZ) implements the sine function. The optical attenuator (Att) is used to set the feedback gain α of the system (see figure 1(a)). The fibre spool gives a round trip time of 7.94 µs. The resistive combiner (Comb) sums the electrical feedback signal with the input signal from the FPGA.
The input and readout layers are programmed on a Xilinx ML605 board, powered by a Virtex 6 FPGA chip. The board is paired with a 4DSP FMC151 daughter card, containing two analog-to-digital converters (ADCs) and two digital-to-analog converters (DACs). For precise synchronisation of the FPGA with the experiment, it is driven by an external clock, generated by a Hewlett Packard 8648A signal generator. The design uses two Galois linear feedback shift registers with a total period of about 10^9 to generate pseudo-random symbols $d(n)$. These are used to compute the inputs $u(n)$ to the reservoir (see below). The inputs $u(n)$ are multiplied by the input mask M_i and sent to the opto-electronic reservoir through the DAC. The resulting reservoir states are collected and digitised by the ADC, and used to compute readout weights W_i and the output signal $y(n)$. The latter is compared to the input signal $d(n)$ and the resulting SER is transmitted to the computer.

Results

In wireless communications, the properties of a channel are highly influenced by the environment. For better equalisation performance, it is crucial to be able to detect significant

channel variations and adjust the RC readout weights in real time. The increasing demand for higher bandwidths in wireless communications will probably require fast and efficient analogue equalisers.
The channel is modelled by the following equations [1, 14]:

$$\begin{aligned} q(n) &= 0.08d(n+2) - 0.120d(n+1) + d(n) + 0.18d(n-1) \\ &\quad - 0.10d(n-2) + 0.091d(n-3) - 0.05d(n-4) + 0.04d(n-5) \\ &\quad + 0.03d(n-6) + 0.010d(n-7), \end{aligned} \tag{2}$$

$$u(n) = p_1 q(n) + 0.036q^2(n) - 0.011q^3(n) + \nu(n), \tag{3}$$

where $\nu(n)$ is the noise. The reservoir computer has to recover the clean input signal $d(n) \in \{-3, -1, 1, 3\}$ from the noisy distorted output signal $u(n)$. The performance is measured in terms of wrongly reconstructed symbols, given by the Symbol Error Rate (SER). The $p_1 \in [0, 1]$ parameter is used to tweak the nonlinearity of the channel. In addition to a constant channel (with $p_1 = 1$), we consider here a simple example of a "switching" channel by changing regularly p_1 between three predefined values $p_1 \in \{0.60, 0.80, 1.00\}$.
Figure 2(a) presents the performance of our reservoir computer, evaluated over one million input symbols, for different signal-to-noise ratios of the wireless channel (green squares). Each value is an average of the lowest rates obtained with different random input masks, and the error bars show the variations of the performance with these masks. We compare our results to those reported in [3], obtained with the same opto-electronic reservoir, trained offline (blue dots). To the best of our knowledge, a SER of 5.71×10^{-6} at 32 dB SNR is the lowest error rate ever reported with an experimental reservoir computer on this task.
Figure 2(b) shows the error rate produced by our experiment in case of a switching noiseless communication channel. Every switch is followed by a steep increase of the SER, as the reservoir computer is no longer optimised for the channel it is equalising. The performance degradation is detected by the algorithm, causing the learning rate λ to be reset to the initial value λ_0. The readout weights are then retrained to new optimal values, and the SER rate correspondingly goes down. We thus demonstrated that online-trained experimental reservoir computers are well suited for dealing with problems that change in real time.

References

[1] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, pp. 78–80, 2004.

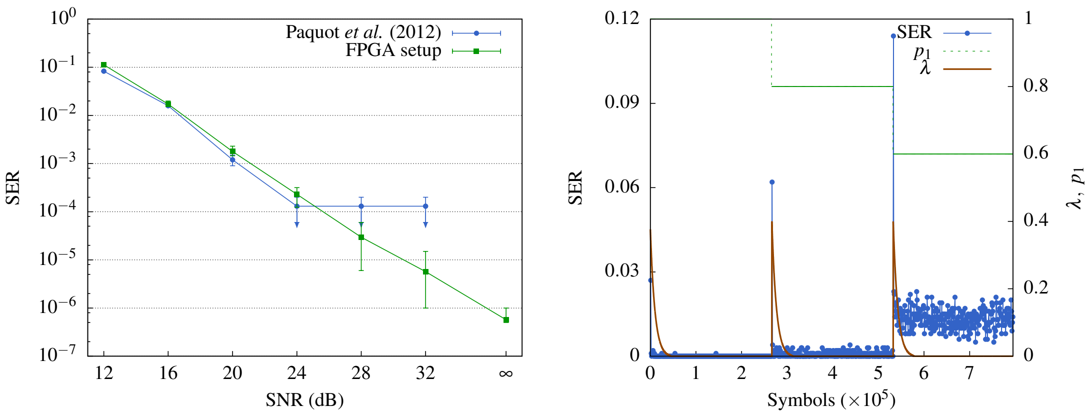
[2] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Comp. Sci. Rev.*, vol. 3, pp. 127–149, 2009.

[3] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar, "Optoelectronic reservoir computing," *Sci. Rep.*, vol. 2, p. 287, 2012.

[4] L. Larger, M. Soriano, D. Brunner, L. Appeltant, J. M. Gutiérrez, L. Pesquera, C. R. Mirasso, and I. Fischer, "Photonic information processing beyond Turing: an optoelectronic implementation of reservoir computing," *Opt. Express*, vol. 20, pp. 3241–3249, 2012.

[5] R. Martinenghi, S. Rybalko, M. Jacquot, Y. K. Chembo, and L. Larger, "Photonic nonlinear transient computing with multiple-delay wavelength dynamics," *Phys. Rev. Lett.*, vol. 108, p. 244101, 2012.

[6] F. Duport, B. Schneider, A. Smerieri, M. Haelterman, and S. Massar, "All-optical reservoir computing," *Opt. Express*, vol. 20, pp. 22783–22795, 2012.



(a) (b)

Figure 2: **(a)** Comparison of experimental results for nonlinear channel equalisation. **(b)** Experimental results for switching channel task. After each switch, the SER goes up, the learning rate λ is reset to λ_0, and then decreases while the SER gradually goes back to a low value.

[7] D. Brunner, M. C. Soriano, C. R. Mirasso, and I. Fischer, "Parallel photonic information processing at gigabyte per second data rates using transient states," *Nat. Commun.*, vol. 4, 2012.

[8] K. Vandoorne, P. Mechet, T. Van Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman, "Experimental demonstration of reservoir computing on a silicon photonics chip," *Nat. Commun.*, vol. 5, 2014.

[9] L. Appeltant, M. C. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, and I. Fischer, "Information processing using a single dynamical node as complex system," *Nat. Commun.*, vol. 2, p. 468, 2011.

[10] Q. Vinckier, F. Duport, A. Smerieri, K. Vandoorne, P. Bienstman, M. Haelterman, and S. Massar, "High-performance photonic reservoir computer based on a coherently driven passive cavity," *Optica*, vol. 2, pp. 438–446, May 2015.

[11] L. Bottou, "Online algorithms and stochastic approximations," in *Online Learning and Neural Networks*, Cambridge, UK: Cambridge University Press, 1998.

[12] P. Antonik, A. Smerieri, F. Duport, M. Haelterman, and S. Massar, "FPGA implementation of reservoir computing with online learning," in *24th Belgian-Dutch Conference on Machine Learning*, 2015. `http://homepage.tudelft.nl/19j49/benelearn/papers/Paper_Antonik.pdf`.

[13] P. Antonik, F. Duport, A. Smerieri, M. Hermans, M. Haelterman, and S. Massar, "Online training of an opto-electronic reservoir computer," in *APNNA's 22th International Conference on Neural Information Processing*, to appear, 2015.

[14] V. J. Mathews and J. Lee, "Adaptive algorithms for bilinear filtering," in *SPIE's 1994 International Symposium on Optics, Imaging, and Instrumentation*, pp. 317–327, International Society for Optics and Photonics, 1994.

[15] A. Rodan and P. Tino, "Minimum complexity echo state network," *IEEE Trans. Neural Netw.*, vol. 22, pp. 131–144, 2011.

[16] G. B. Arfken, *Mathematical methods for physicists*. Orlando FL: Academic Press, 1985.

Acknowledgements

We acknowledge financial support by Interuniversity Attraction Poles program of the Belgian Science Policy Office, under grant IAP P7-35 "photonics@be" and by the Fonds de la Recherche Scientifique FRS-FNRS.