Time and frequency multiplexed implementation of a deep Reservoir Computer

E. Picco,[1] A. Lupo, [1] M. Zajnulina,[1] and S. Massar[1]

[1] Laboratorie d'Information Quantique, CP 224, Université libre de Bruxelles, Av. Roosevelt 50, 1050, Bruxelles, Belgium

Reservoir Computing (RC) is a popular class of recurrent neural networks characterized by having most of the connections untuned and only output weights trained. RCs drew attention because of their advantageous training costs and for being easily implementable in unconventional substrates, constituting a possible work-around for the limits of electronics. We demonstrated two schemes for photonic RC, differentiated by the way in which neurons are encoded and connected. Both have been tested on tasks such as chaotic time-series predictions or nonlinear distortion-compensation. The first scheme is based on frequency multiplexing: neurons are encoded in lines of a frequency comb and are mixed by making lines interfere via phase modulation. The second scheme is based on time multiplexing: neurons are encoded in different timeslot of a signal propagating in a fiber spool and the mixing is achieved by making timeslots overlap during propagation. Organizing RCs in "deep" architectures, i.e.: in series, has the effect of increasing the complexity of their dynamics, thus enhancing their effectiveness. We propose a deep-RC architecture based on the concatenation of our two different RC schemes. Preliminary results show a significative performance improvement on benchmark tasks, showing that the deep architecture benefit from being composed of two RCs with different dynamics.

Introduction

Artificial Neural Networks (ANN) are a powerful brain-inspired computational tool setting the benchmark in multiple challenging tasks. Huge ANNs, composed of thousands of neurons, deliver high performances but require power and time expensive tunings [1]. Randomized Neural Networks (RNN) are ANNs partially consisting of fixed and untrained connections. The RNNs approach has two advantages. First, the training procedure is cheaper. Second, as the internal connections are not required to vary, RNNs can be implemented on various physical substrates, beyond digital electronic platforms.

Reservoir Computing (RC) is a recurrent RNN algorithm implemented in various non-conventional substrates [2]. The untrained recurrent network is called "reservoir" and constitutes a dynamical system exhibiting memory about past inputs. The output of the network is a linear combination of the states of the reservoir neurons, in some cases passed through a nonlinear output function.

Among many possible hardware platforms, photonics is highly suitable for the implementation of reservoir computers, allowing to perform high-bandwidth tasks with reduced energy consumption using commercial-grade devices for light manipulation.

We elaborated two schemes for fiber-optics-based photonic RC. The schemes differ by the way in which neurons are encoded in the light. The first system uses Frequency-Multiplexing (FM-RC) to encode neurons in different wavelengths [3]. The second system deploys Time-Multiplexing (TM-RC) to encode the neurons in different timeslots of a light signal propagating through a fiber loop [4].

Empirical analysis reported in literature [5] show that a "deep" RC, a concatenated network consisting of several RCs, may provide advantages not achievable by a simple

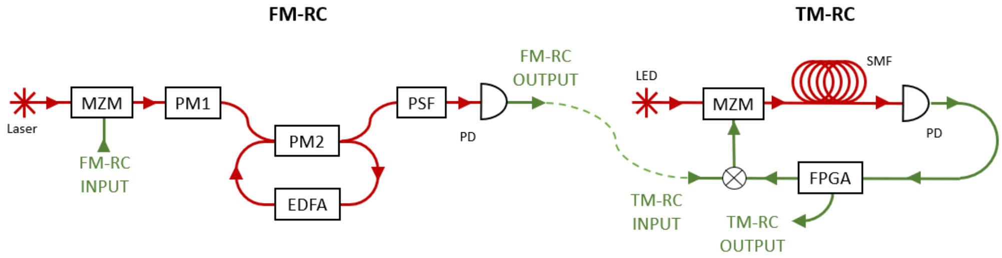

Figure 1. FM-RC: Frequency-Multiplexing Reservoir Computer; TM-RC: Time-Multiplexing Reservoir Computer. MZM: Mach Zehnder Modulator, PM: Phase Modulator; EDFA: Erbium-Doped Fiber Amplifier; PSF: Programmable Spectral Filter; PD: Photodiode; FPGA: Field-Programmable Gate Array. The connections in red are optical, the connections in green are electrical. The FM-RC output drives the TM-RC.

increase of the dimension of one single RC network. Here, we report our first results on a photonic deep-RC obtained by stacking together two physical RC. These preliminary results are a hybrid between experiment and simulation. In particular, the processing through the FM-RC is performed experimentally while the processing through the TM-RC is performed through a realistic simulation. The physical connection between the two RC experiments is feasible and will be the next step of this work.
In the next section, we describe our experimental setups and show how the connection between the two physical RCs is obtained; then we present and discuss our results; finally, in the conclusion, we summarize our findings and perspectives.

Methods

The aim of this work is to explore the concept of concatenating two different photonic RCs to constitute a dual-layer deep-RC. To do so, we employed two experimental systems already presented in the past: a Frequency-Multiplexing RC (FM-RC) [3] and a Time-Multiplexing RC (TM-RC) [4].
In the FM-RC, neurons are encoded in the complex lines of a frequency comb. A monochromatic C-band CW radiation is intensity-modulated according to the input signal by a Mach-Zehnder Modulator (MZM). Afterwards, the radiation is periodically phase-modulated at frequency of $\Omega/2\pi \approx 17GHz$ using a Phase Modulator (PM1). This radio-frequency phase-modulation expands our originally monochromatic radiation into a frequency comb with spectral line separation of Ω and a range that depends on the phase-modulation intensity (we have approximately 25 usable lines). The frequency comb, encoding the input signal, is injected into a fiber loop (polarization maintaining fiber, 15 m long). The loop contains a second Phase Modulator (PM2) and an Erbium-Doped Fiber Amplifier (EDFA). PM2 operates at the same frequency as PM1, generating interference between the lines of the frequency comb which mixes the encoded information. The EDFA partially compensates the losses in the loop, its gain is optimized to obtain the best possible performance. The comb circulating in the fiber loop represents a reservoir state with neurons being encoded in the complex amplitude of the comb lines. The input rate is adjusted to the cavity roundtrip time such that a timestep in the input timeseries fills the cavity. For each roundtrip, the new comb shape (i.e.: the new reservoir state) is determined by the previous state (modified by the passage through PM2) and by the new input. 10% of the radiation is extracted from the loop and directed to a readout circuit composed of a Programmable Spectral Filter (PSF) and a Photodiode (PD). The PSF acts as a tunable band-pass filter allowing to measure independently each comb line to

reconstruct the full evolution of the reservoir. The collected data are later used to train the output weights and produce the overall output of the RC.
In the TM-RC, the neurons are time-multiplexed in the intensity of light traveling through a fiber spool. The reservoir has a ring topology with next-nearest-neighbor coupling. The information flows into an optoelectronic loop composed of the fiber spool and electronic hardware (a Field-Programmable Gate Arrays board, FPGA). The radiation from a Light Emitting Diode (LED) is intensity-modulated via a MZM and injected into a 1.7 km long spool of a Single-Mode Fiber (SMF), connected, on the other end, to a photodiode (PD). The length of the fiber allows for 50 time-multiplexed neurons. The FPGA reads outs the neurons at the PD, mixes them with new inputs, and drives consequently the MZM, which introduces a sine nonlinearity. Also, the FPGA records the full reservoir evolution that is subsequently used to train the output weights and produce the overall output of the RC.
As mentioned, the data processing in the FM-RC was obtained experimentally, while the data processing through the TM-RC was performed in simulation. We use an accurate TM-RC simulation that contains a realistic model for the MZM, PD, and possible desynchronization in the fiber propagation.
The deep-RC scheme that we tested is depicted in Fig. 1. The input signal is supplied to the first reservoir (FM-RC). The FM-RC states, $\boldsymbol{X}_{FM}(t)$, are multiplied by a set of weights $\boldsymbol{W}_{int}$ to obtain a single-dimensional timeseries: $u_{int}(t) = \boldsymbol{W}_{int} \cdot \boldsymbol{X}_{FM}(t)$. The timeseries $u_{int}(t)$ is then injected as an input in the second reservoir (TM-RC). The states of the first RC layer, $\boldsymbol{X}_{FM}(t)$, and the states of the second RC layer, $\boldsymbol{X}_{TM}(t)$, are then concatenated, $\boldsymbol{X}(t) = [\boldsymbol{X}_{FM}(t), \boldsymbol{X}_{TM}(t)]$, such that the vector $\boldsymbol{X}(t)$ contains the value of the neurons of both reservoirs at the timestep t. We use these "total reservoir" states to train the set of output weights $\boldsymbol{W}_{out}$ by means of a ridge regression model.
The choice of $\boldsymbol{W}_{int}$ that represent the connection between the reservoirs is crucial. We tested three possible choices: a random set of weights, a set of weights such that the first layer output already approximates the target, and a set of weights learned via a genetic optimization algorithm (CMA-ES).
The performance of the system is evaluated on a benchmark task consisting in compensating the distortion induced by a nonlinear noisy channel in a simulated communication system (cf. [3]). The Signal-to-Noise Ratio (SNR) can be tuned, thus varying the difficulty of the task.
For each run, 45000 distorted symbols are injected in the deep-RC. The first 1000 outputs are discarded, to avoid evaluating the system during the warm-up transient state. Then, 12000 symbols are employed for training W_{out}, while the remaining 32000 are used for testing. The score is provided in terms of the Symbol-Error Rate (SNR) that denotes the rate of incorrect symbols received. To improve the statistics, the performance values are evaluated as an average over 1000 random repartitions in training and testing set.

Results and discussion

We analyzed the performance of a dual-layer deep-RC using the nonlinear channel equalization as a task. In particular, we tested three different ways of choosing a suitable set of weights $\boldsymbol{W}_{int}$ that connect the first reservoir network layer with the second one. For each technique, we measured the Symbol-Error Rate for different values of the Signal-to-Noise Ratio (SNR) in the simulated channel. The results are shown in Fig. 3.
We first note that the noise in the channel is, in general, detrimental. Hence, for low SNR values the performances are tendentially worse (high SER). For low SNR, the system performance tends to saturate at a minimum SER value. This is because the results depend

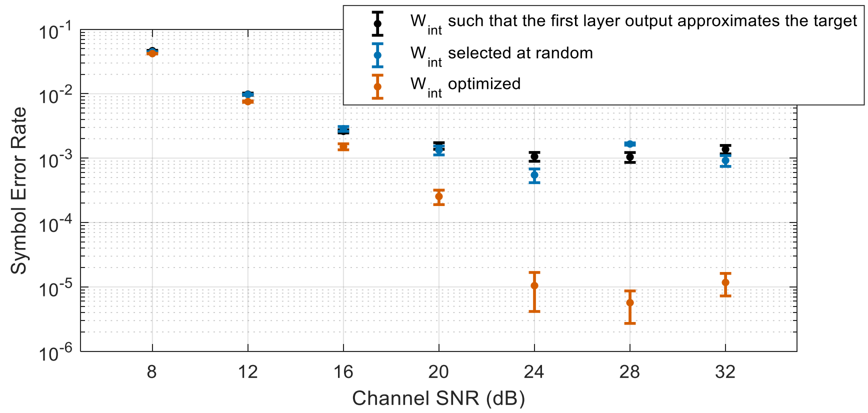

Figure 2. Symbol Error Rate (SER) measured varying the amount of noise in the channel, for three different choices of $\boldsymbol{W}_{int}$.

uniquely on the capability of the system to correct for nonlinear distortions once the noise becomes negligible. The biggest difference between the three ways of choosing $\boldsymbol{W}_{int}$ can be observed in the low-noise regime.

Training $\boldsymbol{W}_{int}$ in such a way that the first layer output approximates the target gives approximately the same result of a random choice for $\boldsymbol{W}_{int}$. Instead, when $\boldsymbol{W}_{int}$ is properly optimized through the CMA-ES algorithm, improvements of up to two orders of magnitude in SER can be observed.

Conclusion

We presented a preliminary study for a scheme of a photonic deep-RC. We showed that stacking two different RCs generates an increase in the overall performance on an idealized channel equalization task. We experimented with different ways of connecting both RC layers observing that a black-box optimization algorithm (in our case, CMA-ES) is necessary to exploit the full power of the deep-RC.

At present, the first layer of the deep-RC is executed in a physical experiment, while the second layer is executed in a realistic simulation.

The positive results obtained in the present work motivate us to pursue in future work a fully experimental implementation of the scheme. Future work will also investigate other architectures, such as stacking two or more layers of the same kind of reservoirs, instead of using heterogeneous ones as in the present work.

Acknowledgement

The authors acknowledge financial support from the European Commission grant 860360-POSTDIGITAL.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature vol. 521, p. 436, 2015.
[2] K. Nakajima, “Physical reservoir computing—an introductory perspective,” Jpn J Appl Phys, vol. 59, issue 6, p. 60501, 2020.
[3] L. Butschek, A. Akrout, E. Dimitriadou, A. Lupo, M. Haelterman, and S. Massar, “Photonic reservoir computer based on frequency multiplexing,” Opt. Lett., vol. 47, issue 4, p. 782, 2022.
[4] P. Antonik, M. Haelterman, and S. Massar, “Brain-Inspired Photonic Signal Processor for Generating Periodic Patterns and Emulating Chaotic Systems,” Phys. Rev. Appl., vol. 7, issue 5, 2017.
[5] M. Freiberger, P. Bienstman, and J. Dambre, “Towards Deep Physical Reservoir Computing Through Automatic Task Decomposition And Mapping,” arXiv:1910.13332, 2019.