A power-efficient architecture for silicon photonic reservoir computing

Stijn Sackesyn,[1,2] Chonghuai Ma,[1,2] Andrew Katumba,[1,2] Florian Denis-le Coarer, [3] Joni Dambre[4] and Peter Bienstman[1,2]

[1] Photonics Research Group, Department of Information Technology, Ghent University - imec, Belgium
[2] Center for Nano- and Biophotonics (NB-Photonics), Ghent University
[3] OPTEL Research Group, LMOPS laboratory, CentraleSupélec,
Université de Paris-Saclay and Université de Lorraine, Metz, France
[4] IDLab, Department of Information Technology, Ghent University - imec, Belgium

Reservoir computing is a neuromorphic computing paradigm which is well suited for hardware implementations. In this work, we present first experimental results on an enhanced reservoir architecture. This architecture has lower optical losses and improved mixing behavior compared to the architectures used in previous silicon photonic reservoir computing designs.

Introduction

Reservoir computing is a machine learning technique in which a nonlinear dynamical system which is also called the reservoir is used for computation. While it was originally implemented as an efficient way to train a neural network [1], it has now grown to a method which is commonly used for classification and regression tasks. Unlike other techniques which optimize the recurrent neural network (RNN) itself to solve a task, the RNN is not modified during training in reservoir computing. Instead, a linear combination is used to train an optimal classifier in the high dynamical state space to which the signal is projected by propagation through the reservoir. By keeping the recurrent network unchanged and train only on the level of the linear output layer makes reservoir computing a computationally cheap method.

Integrated photonic reservoir computing

Although computer science was the original field in which reservoir computing was used as a cheap way to optimize a neural network, it is perfectly suited to be implemented in various hardware platforms. Those dedicated implementations do not suffer from classical digital computer bottlenecks which occur for classical software implementations of reservoir computing, but are specifically designed to operate neuromorphic computing schemes. One such hardware implementation that is especially suited for reservoir computing is silicon photonics. Silicon photonics is a CMOS-compatible platform in which light can be guided, split and combined on a silicon chip. As the reservoir itself is not changed in reservoir computing, a passive network of splitters and combiners, connected using waveguides, is perfectly suited to embody the physical reservoir. Linearly combining the signal at readout nodes can in principle be done fully integrated, but in this work we still made use of grating couplers which couple the light out to an external photodiode. The post processing is done afterwards on a classical computer.

Silicon photonic reservoir computing approaches with a waveguide-based approach typically employ the swirl architecture [2, 3]. This architecture was initially defined in [4] and illustrated in figure 1. The swirl architecture is designed as a RNN and can thus be used as a reservoir but has some shortcomings in terms of losses and mixing, fundamental for the silicon photonics platform: Modal radiation induces losses at each 2

x 1 combiner (for example node 7). These losses are inherent for non-symmetrical reciprocal splitting devices as on average there is a 50% modal mismatch between the two input channels. In terms of mixing, some nodes are partially withdrawn from the dynamics, only consisting of one input and one output (the corner nodes). In general, the most interesting behavior in swirl reservoirs will be at the center, by definition of the architecture, while it could be beneficial to also bring the outer layers more into play.

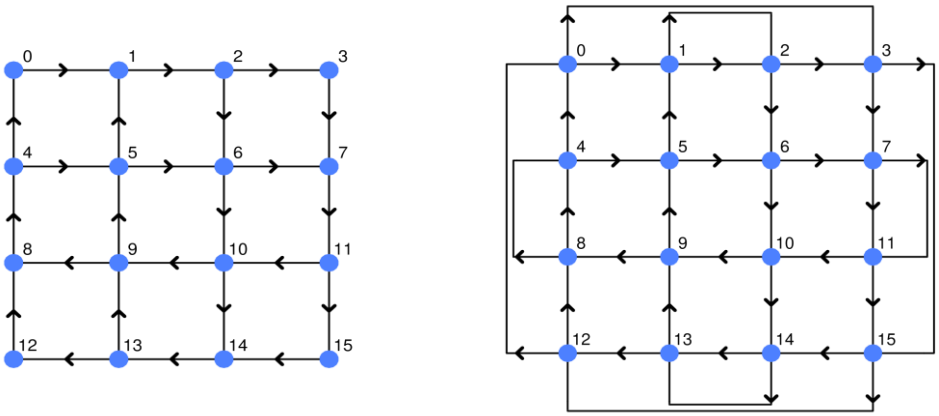

Figure 1: Reservoir architectures on a topological level. Left: SWIRL architecture. Right: Four-port architecture.

In this work, we used the four-port architecture [5] instead of the swirl architecture, which reduces the problems concerning excessive losses by avoiding 2 x 1 devices and by only employing 2 x 2 devices instead. This topology is schematically illustrated in Figure 1.

Fiber distortion compensation

An application for which our approach is very well suited is distortion compensation in optical communication links. The technology which is used commercially in typical systems nowadays is the digital signal processor (DSP). This is a dedicated digital electronics chip. Its main disadvantages are speed limitations for future increasing data rates, high power consumption and latency due to processing. The concept of photonic reservoir computing approach is a future proof competitor as it is faster than classical electronics, has virtually no latency and is orders of magnitude more power efficient.

In this work, a proof-of-concept experiment for distortion compensation was carried out using a 25 km standard single mode optical fiber (SSMF), together with a nonlinear amplifier (EDFA) close to saturation, to enhance the nonlinear effects in the system and thus make the task more challenging. A scheme of the measurement setup is shown in figure 2.

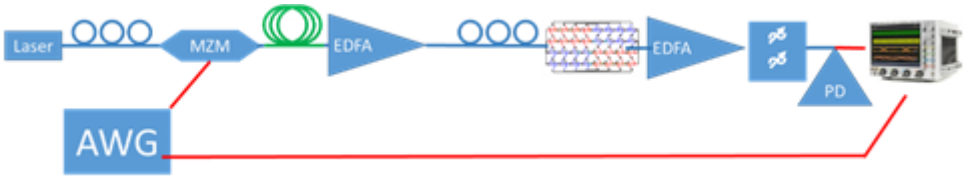

Figure 2: Schematic of the measurement setup. Components from left to right: laser, polarization controller, modulator (32GBPS), optical fiber (SSMF), amplifier (EDFA), polarization controller, photonic reservoir chip, amplifier (EDFA), tunable filter, photo*detector*, real time oscilloscope (RTO). Arbitrary waveform generator (AWG).

The performance of our system was compared to a linear feedforward equalizer for which time-shifted copies of the distorted signal are linearly combined to compensate for fiber distortion effects.

The BER we get for this task using the reservoir computing approach is lower than 1e-4. If we compare this to the tapped filter reference which reaches a BER of order of magnitude 1e-1, we can conclude that the reservoir computing approach indeed performs orders of magnitude better on this task. This finding is visualized in figure 3, where we compare an eye-diagram for both cases.

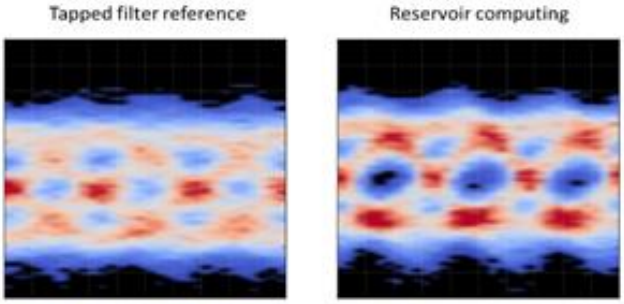

Figure 3: Visual comparison of the eye-diagram for both cases. (left: tapped filter, right: reservoir computing)

Acknowledgements. This work was supported in part by the EU project PHRESCO H2020-ICT-2015-688579 and in part by the Research Foundation Flanders (FWO) under Grant 1S32818N.

References

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

[2] Katumba, A., Freiberger, M., Bienstman, P., Dambre, J.: A Multiple-Input Strategy to Efficient Integrated Photonic Reservoir Computing. Cognitive Computation 9(3), 307–314 (2017).

[3] Vandoorne, K., Mechet, P., Van Vaerenbergh, T., Fiers, M., Morthier, G.,Verstraeten, D., Schrauwen, B., Dambre, J., Bienstman, P.: Experimental demonstration of reservoir computing on a silicon photonics chip. Nature Communications 5, 1–6 (2014).

[4] Vandoorne, K., Dambre, J., Verstraeten, D., Schrauwen, B., Bienstman, P.: Parallel reservoir computing using optical amplifiers. IEEE Transactions on Neural Networks 22(9), 1469–1481 (2011).

[5] Sackesyn S., Ma C., Dambre J., Bienstman P.: An enhanced architecture for silicon photonic reservoir computing. Cognitive Computing 2018 - Merging Concepts with Hardware (2018), pp. 1–2.