2012 International Symposium on Nonlinear Theory and its Applications
NOLTA2012, Palma, Majorca, Spain, October 22-26, 2012

Reservoir computing on nanophotonic chips

Peter Bienstman[†], Kristof Vandoorne[†], Thomas Van Vaerenbergh[†], Martin Fiers[†],
Bendix Schneider[†], David Verstraeten[‡], Benjamin Schrauwen[‡] and Joni Dambre[‡]

†Photonics Research Group, NB-Photonics
Department of Information Technology
Ghent University - imec
9000 Gent, Belgium
Email: Peter.Bienstman@UGent.be

‡ Department of Electronics and Information Systems
Ghent University
Sint-Pietersnieuwstraat 41
9000 Gent, Belgium

Abstract– Reservoir computing is a decade old framework from the field of machine learning to use and train recurrent neural networks and it splits the network in a reservoir that does the computation and a simple readout function. This technique has been among the state-of-the-art for a broad class of classification and recognition problems such as time series prediction, speech recognition and robot control. However, so far implementations have been mainly software based, while a hardware implementation offers the promise of being low-power and fast. Despite essential differences between classical software implementation and a network of semiconductor optical amplifiers, we will show that photonic reservoirs can achieve an even better performance on a benchmark isolated digit recognition task, if the interconnection delay is optimized and the phase can be controlled. In this paper we will discuss the essential parameters needed to create an optimal photonic reservoir designed for a certain task. This design can lead to an efficient implementation of a photonic reservoir on a nanophotonic chip.

1. Introduction

Reservoir Computing (RC) is a training concept for Recurrent Neural Networks (RNNs), introduced a decade ago [1][2]. It comes from the field of machine learning where systems are trained based on examples, instead of programmed with algorithms. In RC a randomly initialized RNN, called the *reservoir*, is used and left untrained. The states of all the nodes of the RNN are then fed into a linear readout, which can then be trained with simple and well established methods. Usually, a mere linear regression is used. Hence, the difficulties of training a recurrent network, such as slow convergence, are avoided as only the readout is changed. Reservoir computing equals or outperforms other state-of-the-art techniques for several complex machine learning tasks. An example is the prediction of the Mackey-Glass chaotic time series several of orders of magnitude better than classic methods [1]. Although the reservoir itself remains untrained, its performance depends drastically on its dynamical regime, determined by the gain and loss in the network. Optimal performance is usually obtained near the edge of stability, i.e., the region in between stable and unstable or chaotic behavior. Hence, to obtain good performance, we need to be able to tune a reservoir's dynamic regime to this edge-of-stability.

A common measure for the dynamic regime is the *spectral radius*, the largest eigenvalue of the system's Jacobian, calculated at its maximal gain state (for classical hyperbolic tangent reservoirs, this corresponds to the largest eigenvalue of the network's interconnection weight matrix). The spectral radius is an indication of the stability of the network. If its value is larger than unity, the network might become unstable. Tuning the spectral radius close to unity often yields reservoirs with close to optimal performance.

2. Photonic Reservoir Computing

Most reported results on reservoir computing use a (randomized) network of hyperbolic tangent or spiking neurons and most have been software based; hence the pursuit of finding a suitable hardware platform for performing the reservoir calculation. This transition offers the potential for huge power consumption savings and speed enhancement. Photonics is an interesting candidate technology for building reservoirs, because it offers a range of different nonlinear interactions working on different timescales. Semiconductor Optical Amplfiers (SOAs), with a saturation of gain and output power, are the optical device closest to the hyperbolic tangent functions used in many RC implementations. This is the reason we chose them as a first medium to verify the usefulness of photonic reservoirs. The SOA model we used is one proposed by Agrawal [3]. It captures the most important features such as gain saturation, carrier lifetime and phase shift depending on the gain.

3. Speech recognition

Speech recognition is a very difficult problem to solve but reservoir computing with classical neural networks has been employed successfully for speech recognition [4]. The task used in this paper is the discrimination between spoken digits, the words ’zero’ to ’nine’, uttered by 5 female speakers.

The dataset and the simulation framework for classical reservoirs are publicly available (http://snn.elis.ugent.be/rctoolbox). As is standard for speech recognition, some pre-processing of the raw speech signal is performed before it is fed into the reservoir. We used the Lyon ear model which is based on the cochlea and highlights certain frequencies typical for our ear [5].

We added babble noise from the NOISEX database, with a Signal-to-Noise Ratio (SNR) of 3 dB (http://spib.rice.edu/spib/selectnoise.html) to increase the complexity of the task. The performance is measured with the Word Error Rate (WER), which is the ratio of incorrect classified samples and the total number of samples.

4. Delay and coherence

In our experiments the input consists of 77 channels, coming out of the Lyon model. With such high-dimensional input, the number of nodes needs to be sufficiently large. Therefore all the experiments were done with a network of 81 (9×9) nodes in a swirl topology (Figure 1). All the connections are nearest neighbor connections and this topology can be easily enlarged, while keeping the length of all connections equal.

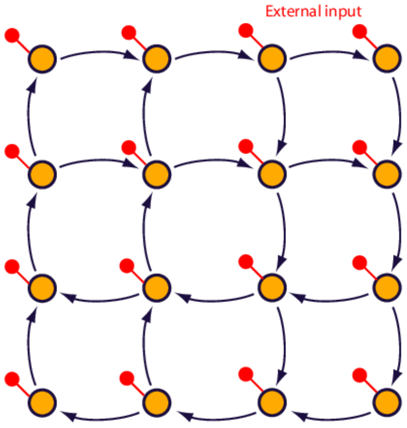

Fig. 1. The swirl topology used in our simulations, here shown in a 4 × 4 configuration

In the experiments we always swept two variables: the phase change and attenuation in every connection. The attenuation influences the spectral radius as it affects the total loss in the reservoir. Because we work with complex amplitudes and to incorporate the influence of coherence, the spectral radius has to be calculated from the complex interconnection matrix, also including the gain in the SOAs. An example of such a sweep can be seen in Figure 2.

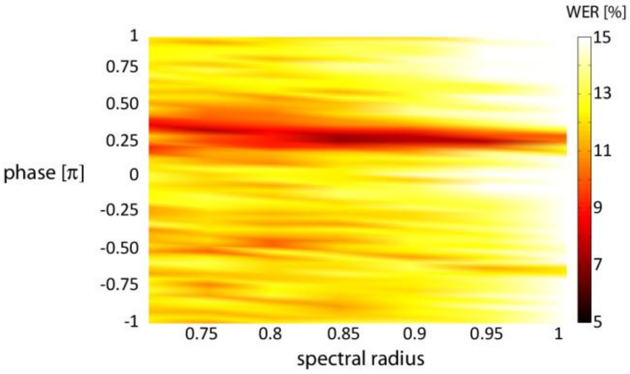

Fig. 2. WER for swirl SOA reservoirs with coherent light and a delay of 6.25 ps where the spectral radius is changed through the attenuation in all the connections. Here, phase is a more critical parameter than spectral radius, as good performance is only found for a narrow band in phase space.

In this case the interconnection delay was very short (6.25 ps) and the performance very phase sensitive. When we change a design parameter, e.g. the delay in the interconnections, such a sweep is done for all the values of that parameter. When we take the best value for every sweep, we can summarize the results as in Figure 3. It shows that there exists an optimal delay in the network.

In a previous paper we have shown that despite several differences between photonic and classical reservoirs (e.g., topology constraints, complex-valued signals and interconnection delays), the use of coherent light in a well-tuned SOA reservoir architecture offers significant performance benefits [6]. The most important design parameters are the delay and the phase shift in the system’s physical connections and with optimized values for these parameters, coherent SOA reservoirs can achieve better results than traditional simulated tanh reservoirs. For longer delays the results become also less phase sensitive.

5. Delay and signal speed

Reservoir memory is related to the typical time scales of the reservoir itself. Therefore, to achieve optimal memory in a reservoir, the relevant time scales of the input signals must be adapted to those of the physical reservoir implementation. Audio signals are rather slow, so we accelerated them to accord with timescales typical for the delays in a network of SOAs (duration of one digit then became in the order of a few hundred ps).

It is, however, interesting to know whether our delay results depend completely on the speech signal itself. In that case we expect the optimal delay to shift according to the input signal speed. This is actually what happens as can be seen in Figure 3 where the optimal results in function of the interconnection delay are shown for rates two times slower and two times faster than our previous experiments. The (a) part shows that, as the speed increases or decreases, the optimal delay shifts just as much. In (b) we have rescaled the X-axis for the different curves the same way as their input speed, so their X-axes all match that of the original one (solid curve–triangles). Here the similarity between the three graphs implies that the optimal delay is indeed a feature of the audio signals themselves as it shifts according to the input rate.

This also means that, confronted with a hardware implementation of a photonic reservoir with certain delays, we can change the input rate to match the delays. For slower signals there is no reason why this should break down, but there are practical limitations to delay lengths on a chip since losses increase with length and they consume area, although recently advances have been made on the Silicon-on-Insulator platform [7]. For ever faster signals at some point the delay in the SOAs will start dominating the interconnection delays.

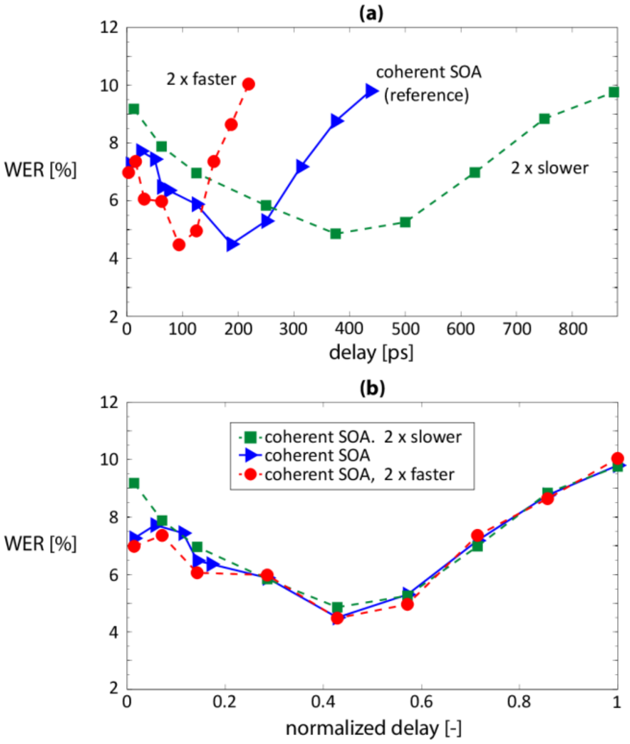

Fig. 3. (a) Results for different speeds of the input signals fed into a coherent swirl SOA reservoir, (b) the same results but plotted on top of each other by changing the X-axis of the different curves the same way as their input rate.

5. Conclusions

We have shown with an isolated digit recognition task that a network of SOAs can be used as a reservoir for reservoir computing and we identified delay as an important design parameter and showed that its value depends on the input speed of the speech signal. This means that for an existing hardware implementation that the input signal should be matched to the hardware delays or vice versa that those delays should be chosen carefully when a photonic hardware reservoir is designed for a specific task.

Acknowledgment

This work has been carried out in the framework of the IAP project Photonics@be of the Belgian Science Policy and the ERC project NaResCo.

References

[1] H. Jaeger and H. Haas, “Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication,” *Science*, vol. 304, no. 5667, pp. 78–80, Apr. 2004.
[2] W. Maass, T. Natschlaeger, and H. Markram, “Real-time computing a without stable states: A new framework for neural computation based on perturbations,” *Neural Computation*, vol. 14, no. 11, pp. 2531–2560, 2002.
[3] G. P. Agrawal and N. A. Olsson, “Self-phase modulation and spectral broadening of optical pulses in semiconductor-laser amplifiers,” *IEEE Journal of Quantum Electronics*, vol. 25, no. 11, pp. 2297–2306, 1989.
[4] D. Verstraeten, B. Schrauwen, D. Stroobandt, and J. V. Campenhout, “Isolated word recognition with the liquid state machine: a case study,” *Information Processing Letters*, vol. 95, no. 6, pp. 521–528, 2005.
[5] R. Lyon, “A computational model of filtering, detection, and compression in the cochlea,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), vol. 7, Paris, France, May 1982, pp. 1282–1285.
[6] K. Vandoorne, J. Dambre, D. Verstraeten, B. Schrauwen, and P. Bienstman, “Parallel Reservoir Computing Using Optical Amplifiers,” *IEEE Transactions on Neural Networks*, vol. 22, no. 9, pp. 1469–1481, Sep. 2011.
[7] W. Bogaerts and S. K. Selvaraja, “Compact Single-Mode Silicon Hybrid Rib/Strip Waveguide With Adiabatic Bends,” *IEEE Photonics Journal*, vol. 3, no. 3, pp. 422–432, Jun. 2011