Optical information processing: Advances in nanophotonic reservoir computing
— Source link 🗗

Martin Fiers, Kristof Vandoorne, T. Van Vaerenbergh, Joni Dambre ...+2 more authors

Institutions: Ghent University

Related papers:

- Photonic reservoir computing and information processing with coupled semiconductor optical amplifiers

- Optical signal processing with a network of semiconductor optical amplifiers in the context of photonic reservoir computing

- Photonic reservoir computing: A new approach to optical information processing

- Reservoir computing: a photonic neural network for information processing

- Optoelectronic reservoir computing.

Optical Information Processing:
Advances in Nanophotonic Reservoir Computing

M. Fiers[1,3], K. Vandoorne[1,3], T. Van Vaerenbergh[1,3], J. Dambre[2], B. Schrauwen[2], P. Bienstman[1,3], *Member, IEEE*

[1]*Ghent University, Department of Information Technology, Sint-Pietersnieuwstraat 41, 9000 Ghent, Belgium*
[2]*Ghent University, Department of Electronics and Information Systems, Sint-Pietersnieuwstraat 41, 9000 Ghent, Belgium*
[3]*Center for Nano- and Biophotonics (NB-Photonics), Ghent University, Sint-Pietersnieuwstraat 41, 9000 Ghent, Belgium*
Tel: +32 (0) 9264 3316. Fax: +32 (0) 9264 3593, e-mail: Martin.Fiers@intec.ugent.be

ABSTRACT

We present a complex network of interconnected optical structures for information processing. This network is an implementation of reservoir computing, a novel method in the field of machine learning. Reservoir computing can be used for example in classification problems such as speech and image recognition, or for the generation of arbitrary patterns, tasks which are usually very hard to generalize. A nanophotonic reservoir can be constructed to perform optical signal processing. Previously, simulations demonstrated that a reservoir consisting of Semiconductor Optical Amplifiers (SOA) can outperform traditional software-based reservoirs for a speech task.

Here we propose a network of coupled photonic crystal cavities. Because of the resonating behaviour, a lot of power is stored in the cavity, which gives rise to interesting nonlinear effects. Simulations are done using a novel software tool developed at Ghent University, called Caphe. We train this network of coupled resonators to generate a periodic pattern using a technique called FORCE. It is shown that photonic reservoirs can outperform classical software-based reservoirs on a pattern generation task.

Keywords: nanophotonic reservoir computing, photonic crystal cavities, nonlinear dynamics, coupled resonators, pattern generation

1. INTRODUCTION

Machine learning is an information processing paradigm often used in cases where it is difficult to generalize input-output relations, for example in classification problems such as speech and image recognition, or for the generation of patterns. A recently proposed methodology in this field is reservoir computing [1]-[2] , which has the advantage that the readout mechanism is very simple to train. Until now, they have been used mainly in a software implementation, i.e. the neurons are modelled in software. These are however limited in speed and power efficiency. Nanophotonic reservoir computing provides an excellent platform for a hardware implementation: First because of the presence of unique nonlinear effects that enrich the dynamics of the system. Second, because the signals have a phase in addition to the amplitude, which add additional degrees of freedom as opposed to purely electronic hardware implementations and third, the very high speeds achievable in optics is an important advantage over electronic reservoirs.

One of the previously proposed photonic implementations employs Semiconductor Optical Amplifiers (SOA) as the basic building blocks for the reservoir. It is shown using simulations that they can outperform traditional software-based hyperbolic tangent reservoirs for a speech task [3].

Here we propose a network of coupled photonic crystal cavities to generate a coherent pattern using a technique called FORCE [4]. Because of the resonating behaviour, a lot of power is stored in the cavity, which gives rise to interesting nonlinear effects such as the Kerr-effect, free carriers and temperature. When coupling two or more resonators, the Kerr-effect gives rise to self-pulsation and chaos [5].

Simulations are done using a novel software tool developed at Ghent University, called Caphe [6]. Using the same tool, spiking neurons are investigated. These are usually more difficult to model in software (in terms of speed and stability), but spiking neural networks have more realistic properties compared to analog neural networks. In this paper, first we give a brief introduction of reservoir computing, and show how we can train a reservoir to generate periodic patterns using the technique called FORCE. Then we explain how we construct the optical reservoir, and explain what the restrictions are concerning the topology. After this, we compare the discrete tangent hyperbolic reservoir with the optical reservoir by generating a sum of sine waves. This problem is called the Multiple Superimposed Oscillator (MSO) task.

2. RESERVOIR COMPUTING

2.1 Basic principle

There are different types of reservoirs. The two most commonly used types are the echo state networks (ESNs [1]) and liquid state machines (LSM [2]). The first one mostly resembles our architecture, and was originally formulated in discrete time through the reservoir state update function:

$$\mathbf{x}[t + \Delta t] = (1 - \lambda)\mathbf{x}[t] + \lambda\, \mathbf{f}(\boldsymbol{W}_{in}\mathbf{u}[t] + \boldsymbol{W}_{res}\mathbf{x}[t]) \tag{1}$$

Here, $\mathbf{x}[t]$ are the states of the reservoir (size N) and $\mathbf{u}[t]$ is the input of the reservoir at time step t. $\boldsymbol{W}_{in}$ is the weight matrix for all the connections from the reservoir to the reservoir (size N x N), $\boldsymbol{W}_{res}$ is the weight matrix for the connections from the input to the reservoir, and $\mathbf{f}(\mathbf{x})$ is the neuron function. Typically, this is a sigmoid function (e.g. the hyperbolic tangent function, tanh). The parameter λ is called the leak rate of the network. This effectively adds a first-order low-pass filter to each neuron. When $\lambda = 1$, the previous state is just discarded. The closer λ is to zero, the more each neuron remembers its previous state. Especially in temporal tasks (such as pattern generation), it is important to have sufficient memory.

In reservoir computing, the $\boldsymbol{W}_{in}$ matrix is typically chosen random, e.g. sampled from a uniform distribution between [-1..1]. The $\boldsymbol{W}_{res}$ matrix is also chosen randomly, but then rescaled so that the network is on the edge of chaos. Tuning the reservoir close towards the edge of chaos has been found to give optimal results for many tasks. The task of the reservoir is to transform the input signal $\mathbf{u}[t]$ to a higher-dimensional space, where it is easier to extract features which are present in the input signal. Finally, the output of the reservoir is calculated using a linear readout function:

$$\mathbf{z}[t] = \boldsymbol{W}_{out}\mathbf{x}[t] \tag{2}$$

The training is done by first feeding all input $\mathbf{u}[t]$ to the network. Afterwards, the reservoir computing system is trained by adjusting the weights $\boldsymbol{W}_{out}$. Typically this is solved using linear regression. A commonly used scheme for solving this problem is ridge regression. Suppose the target outputs are given by the matrix $\boldsymbol{B}$, then $\boldsymbol{W}_{out}$ is given by:

$$\boldsymbol{W}_{out} = (\boldsymbol{A}^T\boldsymbol{A} + \gamma\boldsymbol{I})^{-1}\boldsymbol{A}^T\boldsymbol{B} \tag{3}$$

Where γ is a regularization constant, which is optimized using so-called cross-validation.

2.2 Generating periodic patterns using FORCE

FORCE stands for First Order Reduced and Corrected Error [4]. It is an on-line learning rule, which means the output signal is fed back to the input and weights are modified during the simulation. This is in contrast to off-line learning rules, where the output weights are trained after feeding all input to the network (see section 2.1). There are different ways to modify the readout weights. One of the most commonly used mechanisms is using the recursive least squares (RLS) rule (in the remainder of this paper, we use only one scalar output, so z[t] is scalar and $\mathbf{w}_{out}$ and $\mathbf{w}_{in}$ are vectors):
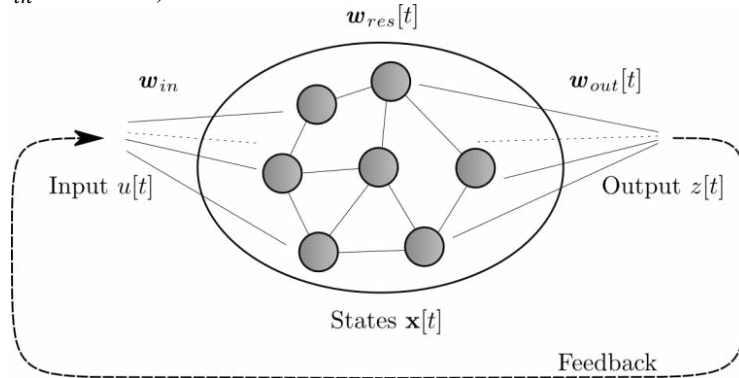

Figure 1. Principle of FORCE: The output of the reservoir is fed back to the input. During training, the output weights $\mathbf{w}_{out}[t]$ are modified using the RLS rule which is described below.

First we calculate the error between the output signal $z[t]$ and the target signal $s[t]$: $e[t] = z[t] - s[t]$. We then define the matrix $\boldsymbol{P}$ as the running estimate of the inverse of the correlation matrix of the network states, plus a regularization term. Its initial value equals $\boldsymbol{P}_0 = \frac{1}{\alpha}\boldsymbol{I}(N, N)$, where α acts as a learning rate. Smaller values of α mean faster learning, but the weight changes can be so rapid that the algorithm becomes unstable. Also, for smaller α, the norm of the weights are typically larger, which is not always wanted. The matrix $\boldsymbol{P}$ is updated by the following equations:

$$\mathbf{k}[t] = \boldsymbol{P}(t - \Delta t)\mathbf{x}[t] \tag{4}$$

$$c = \frac{1}{1 + \mathbf{x}\, P\, \mathbf{x}} \tag{5}$$

$$P[t] = P(t - \Delta t) - \mathbf{k}.(\mathbf{k}^T c) \tag{6}$$

Finally, the output weights are adjusted using the following equation:

$$\mathbf{w}_{out}[t] = \mathbf{w}_{out}[t - \Delta t] - e[t]\mathbf{k}^T c \tag{7}$$

The weights $\mathbf{w}_{out}$ are changing in time. If the algorithm converges, the norm of $\mathbf{w}_{out}$ also converges to a fixed value and the system is on a stable attractor. It is important that this norm does not grow too big: large output weights are an indication that the training is overfitting, which reduces the robustness of the system (for example, when perturbing the output signal the system might jump to another attractor or even become chaotic).

3. PHOTONIC RESERVOIR COMPUTING

To construct a photonic reservoir, we use photonic crystal cavities fabricated using SOI technology [10]. We model these devices using coupled mode theory. The basic equations describing one cavity are the following:

$$\frac{da_j}{dt} = \left[i(\omega_0 + \delta\omega_j) - \frac{1}{\tau_j}\right]a_j + ds_{j,l,in} + ds_{j,r,in} \tag{8}$$

$$s_{j,l,out} = e^{i\varphi}\, s_{j,l,in} + da_j \tag{9}$$

$$s_{j,r,out} = e^{i\varphi}\, s_{j,r,in} + da_j \tag{10}$$

Where $|a_j|^2$ represents the energy in the mode of cavity j. $s_{j,l,in}$ is the input signal on cavity j from left, $s_{j,r,in}$ the input from right. The term $e^{i\varphi}$ is the reflection from the cavity. The nonlinearity is provided by the term $\delta\omega_j = -|a_j|^2/P_0\tau^2$, where P_0 is the characteristic nonlinear power of the cavity [9]. Typically P_0 can be brought below 77 mW [5]. The cavity has a lifetime of $\tau = 1.39$ ps, $\varphi = 0.2\pi$ and $\omega_0 = 1.55\mu m$. These cavities are then connected using an appropriate topology and simulated with Caphe, a circuit simulation tool developed at Ghent University [6].

In a hardware implementation, there are several restrictions w.r.t. the topology. The technology used here is Silicon On Insulator (SOI). This means all components are on one chip, which reduces the footprint drastically compared to bulk photonics. The structure is planar, and too many optical crossings should be avoided because they add a loss term (approximately 0.17 dB per crossing [8]). Another important consideration is the fan-out. The output power of one cavity needs to be divided over the different output ports, and this power drops quickly with increasing fan-out. To keep the input powers about the same order of magnitude, the fan-out should be minimized. Due to reciprocity, the same holds for fan-in: If only one input to a cavity would be activated, still the power that arrives at the cavity is only a fraction of this input power. After taking these restrictions into consideration, we propose the waterfall topology, as displayed in Fig. 2.
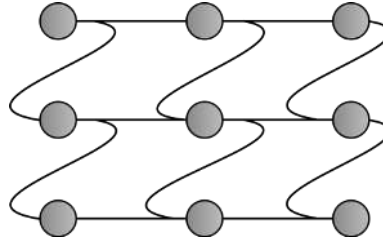

Figure 2. Topology of the photonic reservoir: To avoid optical crossings and to minimize fan-in and fan-out, each photonic crystal cavity (represented by a circle) is connected to its closest neighbours in a 'waterfall' topology.

4. RESULTS

The quality of the reservoir is measured by the ability of the reservoir to follow a target signal $s[t]$ after training this reservoir. A standard benchmarking task in the RC world is the Multiple Superimposed Oscillator (MSO) task. The target signal is given by a sum of two sine waves with different frequencies:

$$s[t] = \sin(\omega_1 t) + \sin(\omega_2 t) \tag{11}$$

For a discrete reservoir, $\omega_1 = 0.2$ and $\omega_2 = 0.311$.

The performance is measured by the normalized root mean square error (NRMSE) between s[t] and z[t]. The period of the first oscillation is $T_1 = 2\pi/\omega_1 \cong 31{,}4s$. First there is a warmup (noise input) of $15.T_1$, then the reservoir is trained for $400.T_1$, then it is left in freerun ($\mathbf{w}_{out}$ fixed) for $2000.T_1$. Then we take the last $100.T_1$ samples, and move them over the first $1000.T_1$ timesteps of the freerun phase, each time calculating the NRMSE. The lowest value in this list of errors is the resulting NRMSE. The system should not diverge from the

attractor and for that reason we simulate a large amount of periods ($2000 \cdot T_1$). When a system is unstable, the calculated NRMSE is very high. By sweeping the last $100 \cdot T_1$ over a large window, frequency drifts are not taken into account. For that we would need another performance measure. The reservoir size is 200, the learning rate $\alpha = 0.01$ and the feedback strength is 1. Fig. 3 shows the simulation results for a discrete reservoir (left) and for an optical reservoir (right).
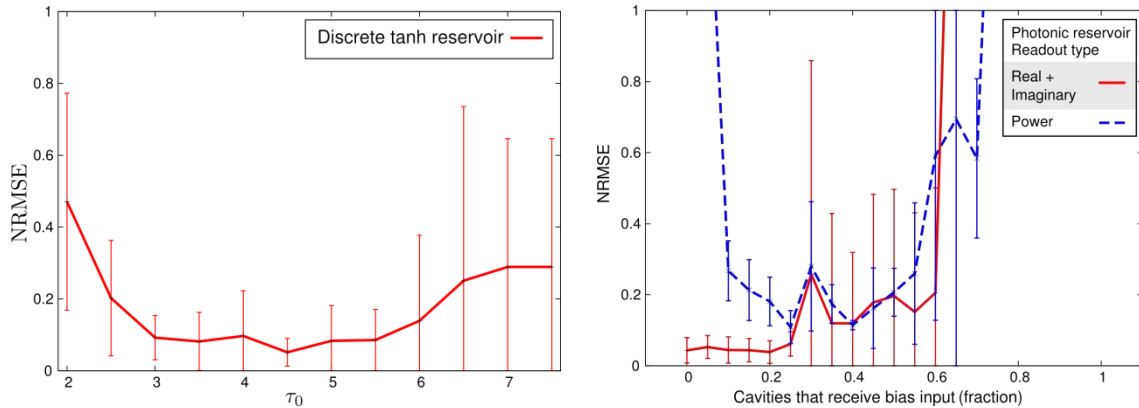

Figure 3. Normalized root mean square error (NRMSE) between the output of the reservoir $z[t]$ and the target signal $s[t]$. Left: Simulation of a standard tangent hyperbolic reservoir for different leak rates ($\tau_0 = 1/\lambda$). Right: Simulation of a photonic reservoir for different input fractions. Two types of readout layers were used: real/imaginary output (full line), only power output (dashed). All simulations were performed 50 times with different initial conditions, and the error bars indicate the standard deviation of the result.

For the discrete reservoir, the spectral radius (defined as the absolute value of the largest eigenvalue of the connection matrix $\boldsymbol{W}_{res}$) is 1.5. We sweep the inverse of the leak rate $\tau_0 = 1/\lambda$. The best NRMSE (averaged over 50 samples) is 0.0515 for $\tau_0 = 4.5$.

For the optical reservoir, we first looked for the optimal signal frequency. This turned out to be approximately $\omega_1 = 0.3 * 2\pi/\tau$ (where τ is the cavity lifetime). We sweep the fraction of cavities that receives an input bias ($P_{bias} = 1.30\, P_0$). For this input power, a series of two coupled resonators self-pulsate [5]. When too many bias input is provided to the reservoir (above 70%), the reservoir fails to generate the requested periodic pattern. The best NRMSE is found to be approximately 0.0387 for an input fraction of 20%. This proves that optical reservoirs can be used to generate periodic patterns, and the performance for the MSO task is better than for the typical tangent hyperbolic reservoir.

5. CONCLUSIONS

We investigated a new type of hardware implementation for reservoir computing using photonic crystal cavities. The simulation results show that the optical reservoir performs better than a standard leakytanh reservoir with optimized parameters for the MSO task. Future research will focus on the experimental readout and feedback mechanism, and on analyzing the variability between the individually fabricated photonic crystal cavities.

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. Natschläger and H. Markram: Rea-time computing without stable states: A new framework for neural computation based on perturbations", *Neural Computation*, Vol. 14, pp. 2531-2560, 2002.
[3]   Vandoorne et al, IEEE Transactions on Neural Networks, 22(9), p.1469-1481, 2011
[4]   D. Sussillo, *et al.*: Generating Coherent Patterns of Activity from Chaotic Neural Networks, *Neuron*, Vol. 63(4), pp 544-557, 2009
[5]   B. Maes, M. Fiers, P. Bienstman: Self-pulsing and chaos in short chains of coupled nonlinear microcavities, *Physical Review A*, vol. 80, p.033805, 2009
[6]   M. Fiers, *et al.*: Time-domain and frequency-domain modeling of nonlinear optical components at the circuit-level using a node-based approach, *J. Opt. Soc. Am. B*, Vol. 29, pp. 896-900, 2012.
[7]   site: www.intec.ugent.be/caphe
[8]   W. Bogaerts, *et al.*: Low-Loss, Low-Crosstalk Crossings for SOI Nanophotonic Waveguides, *Optics Letters*, Vol. 32, pp. 2801-2803, 2007
[9]   M. Soljačic, *et al.*:Optimal bistable switching in nonlinear photonic crystals, *Physical Review E*, Vol. 66, pp. 055601, 2002
[10] W.C.L. Hopman, *et al.*: Nano-mechanical tuning and imaging of a photonic crystal micro-cavity resonance, *Optics Express*, Vol. 14, pp. 8745-8752, 2006