Photonic Reservoir Computing: A New Approach to Optical Information Processing

Kristof Vandoorne[1]*, Martin Fiers[1], David Verstraeten[2], Benjamin Schrauwen[2], Joni Dambre[2] and Peter Bienstman[1]*

[1] *Photonics Research Group, Department of Information Technology, Ghent University – IMEC Sint-Pietersnieuwstraat 41, 9000 Gent, Belgium*

[2] *PARIS, Department of Electronics and Information Systems, Ghent University Sint-Pietersnieuwstraat 41, 9000 Gent, Belgium*

** Tel: +32 9 264 3447, Fax: +32 9 264 3593, e-mail: Kristof.Vandoorne@intec.UGent.be*

ABSTRACT

Despite ever increasing computational power, recognition and classification problems remain challenging to solve. Recently advances have been made by the introduction of the new concept of reservoir computing. This is a methodology coming from the field of machine learning and neural networks and has been successfully used in several pattern classification problems, like speech and image recognition. The implementations have so far been in software, limiting their speed and power efficiency. Photonics could be an excellent platform for a hardware implementation of this concept because of its inherent parallelism and unique nonlinear behaviour.

We propose using a network of coupled Semiconductor Optical Amplifiers (SOA) and show in simulation that it could be used as a reservoir by comparing it on a benchmark speech recognition task to conventional software implementations. In spite of several differences, they perform as good as or better than conventional implementations. Moreover, a photonic implementation offers the promise of massively parallel information processing with low power and high speed.

We will also address the role phase plays on the reservoir performance.

Keywords: photonic reservoir computing, integrated optics, semiconductor optical amplifiers, nonlinear optics, optical neural networks, speech recognition

1. INTRODUCTION

Although computers and algorithms are becoming ever stronger and more powerful, there are problems that are not easily solved with an algorithmic approach. Speech and image recognition are among them, while humans seem to have a natural ability for such tasks. This has inspired the fields of machine learning and Artificial Neural Networks (ANN) where people try to build systems suited for this class of problems and where they often take models of the human brain as an inspiration. ANNs consist of a network of nodes and interconnections between these nodes, just as neurons in our brain are connected to many other neurons. The nodes itself perform a function which can be as easy as applying a tangent hyperbolic or as complex as a model approximating the behaviour of biological neurons.

What sets ANNs apart from algorithms is the fact that they are, just as humans, *trained* to perform a task. The training is done through learning by example. In a simplified manner this training takes the following form. We want the ANN to distinguish between different classes of data (e.g. different words in speech). Examples are obtained and ideally sufficient examples of every class are used. Every example consists of a certain input and a desired output that is expected from the ANN. In our case the learning will be supervised, which means that an external observer is involved in judging the performance of the output of the ANN. Part of the examples is fed to the ANN and the output is monitored. Ideally the output should be the same as the desired output, and therefore the interconnection strength between the nodes (or interconnection weight) is changed until the output of the ANN is as close as possible to the desired output. The second part of the examples, which differ from the examples used for training, is then used to evaluate the trained ANN. If the training was done well, then the ANN should perform as good on the unseen data of the test set. In this case we say that the ANN generalizes well, which means that it has learned to distinguish between the different classes that the examples are samples from and not just between the examples itself.

Feed-forward Neural Networks, which have no feedback and are structured in layers, have been studied for a long time and are used for a number of applications. The lack of feedback makes that there is just a mapping from input to output and several well established training rules for the adaptation of the weights exist. The drawback is that they lack memory needed for many real world applications which have temporal behaviour such as speech recognition. Recurrent Neural Networks (RNN) do have feedback connections but their use remained problematic for a long time due to reasons as slow or non-convergence during training.

Reservoir Computing (RC) is a new training concept for RNNs, introduced a few years ago, that combined the advantages of both recurrent and feed forward neural networks [1,2]. In this framework a RNN is used but left untrained and we will call it the 'reservoir'. The state of all the nodes of the RNN is then fed into a linear readout, which can then be trained with well established methods. Here as well, training means adapting certain connections strengths, but only within the readout, not within the reservoir. The interesting properties of RNNs and its associated memory are maintained, while the training remains easy because linear methods can be used.

Although the reservoir itself remains untrained, this does not imply that any reservoir will do. Rather, from experiments and experience it turns out that most reservoirs perform differently in different dynamical regimes, but best on the edge of stability, i.e. the region in between stable and unstable to chaotic behaviour. This region is determined by the total amount of gain and loss in the network. A measure often used is the spectral radius, the largest eigenvalue of the interconnection matrix. At zero input it is an indication of the stability of the network. If its value is larger than one, the network will be unstable. Intuitively this can be understood because over time this connection matrix will be applied again and again. If one of its eigenvalues is larger than one, then there is gain in the overall network, which will eventually result in unstable behaviour. In classical reservoirs based on tangent hyperbolic functions, the spectral radius is often used as a measure to create a network that is on the edge of stability for good performance, i.e. with a spectral radius just below one. It is important to mention that this measure is just an indication since it is only valid for linear functions and for zero input. For non-zero input and nonlinear functions stable behaviour can also happen for spectral radii larger than one.

2. PHOTONIC RESERVOIR COMPUTING

The idea of RC is actually very broadly applicable and many different types of reservoirs are currently being investigated. One way of interpreting the reservoir and the readout function is to view the reservoir as a special and advanced kind of pre-processing or filtering of the input before the readout function. In this view, the reservoir essentially mixes the inputs together, so that the interesting features are more easily extracted by the readout. A nonlinear mixing often seems to offer advantages over linear mixing when dealing with more complicated problems such as speech recognition.

Most implementations so far 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. What makes a hardware implementation even more attractive is the fact that the computation in RC happens through the transient states and changing dynamical behaviour. This is in sharp contrast with digital computation where the state is only important after the transients have died out.

Photonics seems like a very interesting candidate of building a reservoir, because it has a range of different nonlinear interactions working on different timescales. It also offers the promise of being more power efficient. There remain, however, many challenges as well. If you encode the information in changing power levels, then it becomes difficult to have negative weights and to subtract signals. A topology made on a 2D chip, which is the case for most photonic chips nowadays, limits the freedom in connectivity that exists in software implementations, since one would like to minimise the number of crossings. When using a coherent light source, the amplitude and accompanying phase start to play a role as well, whereas traditional RC is only amplitude based. In this paper we will show through simulation results that despite these limitations, photonic reservoirs can perform quite well on benchmark problems.

3. SIMULATION RESULTS

In this section the speech recognition task that we used as a benchmark problem will be described, as well as the model we used to simulate a photonic reservoir. In our case the reservoir consists of a network of coupled Semiconductor Optical Amplifiers (SOAs).

3.1 Speech Recognition

Speech recognition is a very difficult problem to solve and methods based on ANNs have been among the state of the art for a long time. Reservoir Computing with classical neural networks has been employed with success for speech recognition. The speech recognition that we have used in this paper is about digit recognition, zero to nine, uttered by 5 female speakers ten times. The dataset and simulation framework for classical reservoirs is publicly available and can be found here: 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. Often these methods involve a transformation to the frequency domain and highlighting certain frequencies typical for our ear by using some kind of ear model. The model used for the results in this paper was the Lyon ear model [3].

Audio signals are rather slow and in our simulation we fed the speech signal at much higher speeds to the photonic reservoir, at timescales typical for the delays in a network of SOAs. One of the reasons is that the relation between the time scales of the reservoir and those of the input is important [4]. This made the duration of an average sample in the order of a few hundred ps. So although we use this task to demonstrate the potential of photonic reservoir computing, we don’t propose to use photonic reservoir as a platform for standard real-time, slow audio signals.

3.2 Simulation model

SOAs with their saturation of the gain and output power are the optical device closest to the tangent hyperbolic functions used in many ANN implementations. That 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 [5]. It captures the most important features such as gain saturation, carrier lifetime and phase shift depending on the gain. Spectral hole burning, cross gain and phase modulation were not considered, since the operation is set to be at one wavelength.

A time step based rate equation model, solved by a fourth order Runge-Kutta, of a network of coupled SOAs was then plugged into the freely available RC toolbox mentioned earlier. This toolbox offers a number of reservoir simulations and benchmark problems that can be simulated and tested. Furthermore almost every aspect of it can be changed to make it suitable to specific experiments.

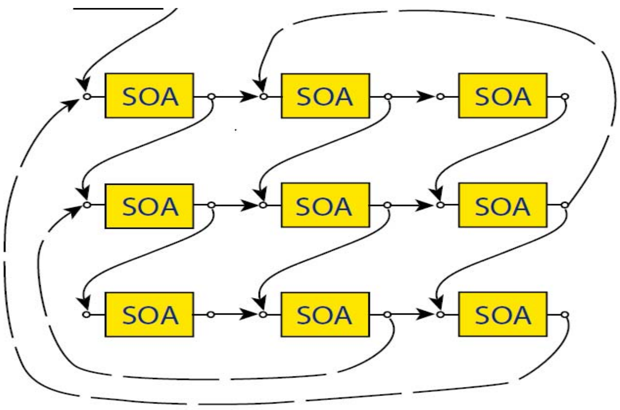

Figure 1. The topology used for the SOA simulations.

3.3 Results

In these experiments the input consists of 77 channels at every time step. These channels are the result of the pre-processing of the speech data. Therefore we have chosen are network large enough. All the experiments were done with a network of 81 nodes. An example of the topology used can be seen in Fig. 1. There a network of 3 by 3 SOAs is shown. The connections are made in such a way that they don't cross. The information flows from the top left to the bottom right SOAs with nearest neighbour connections. The feedback is assured by having as many feedback connections on the edges of the network as possible without having to use crossings. The topology we used was a 9 by 9 network but the construction method was the same.

In the experiment we varied two variables: the phase change and attenuation in every connection. Although in practice these wouldn't be the variables that are swept, they are orthogonal and therefore provide an interesting insight in the behaviour of the network. In reality the input current of the SOAs and the wavelength of the light can be used, but since the input current of the SOAs influences the gain, which in turns influences the amount of phase change inside an SOA, these variables are not orthogonal.

The total amount of gain and loss in the network is also calculated by means of the spectral radius, mentioned earlier. Since coherent light is used, the signals can be represented by complex amplitudes. It is therefore important that the weights in the interconnection matrix, used for the calculation of the spectral radius, are the weights for these complex signals and not for their intensity. In this way interference effect are accounted for.

An example of a result of such an experiment is shown in Fig. 2. Here a clear transition at a spectral radius around 1 can be seen. Above 1 the results become suddenly a lot worse and this is due to the fact that instability kicks in because there is gain in the network. This regime actually corresponds with one where some SOAs would become lasers. Our simulation model is probably not suited for addressing a network of coupled lasers, so it is not possible to state with certainty that this regime is very bad for RC. Maybe some kind of emergent behaviour, useful for RC, could arise from coupled lasers, but this is the topic of ongoing investigation.

Another consequence of using coherent light is that the delay in the connections changes the phase of the light according to the wavelength, length and the effective index of the connection. All the connections were considered equally long in this experiment, which means that the phase change in every connection can be changed at the same rate, for example by changing the wavelength. Phase is important, since it determines the interference of the light when it combines in front of every SOA. Looking at Fig. 2, it becomes clear that the network performs better for some phase changes or interference. The optimal result for this task was a Word Error Rate (WER), which is the percentage of the words incorrectly recognized, of about 1%. This is comparable to the results achieved by classical tangent hyperbolic networks, although the performance of both software and photonic reservoirs could be improved when a low-pass filter is added to all of the nodes.

From Fig. 2 it becomes apparent that the phase is very important to the performance of the reservoir. Phase is much more sensitive than gain, so it is important go have an optimal region that is as vast as possible for phase.

This can be understood because for a small optimal region, a small phase change can be enough to lose your optimal performance. A lot of ongoing investigation is about finding networks with large optimal areas (making them phase 'independent'). An easy way of doing this is working with incoherent light, but the results are typical a lot worse for incoherent networks.

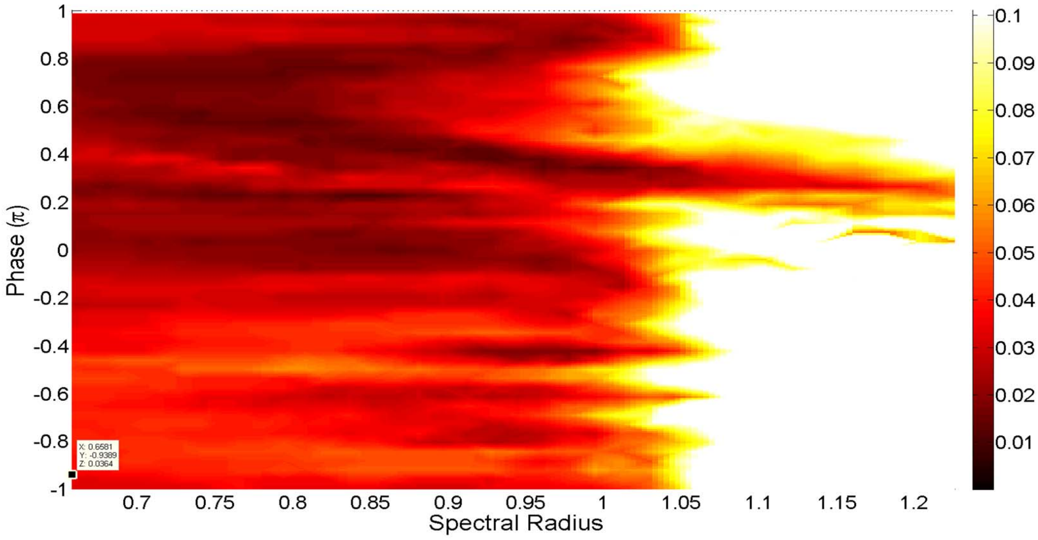

Figure 2. Simulation result for a network of coupled SOAs for speech recognition. The x-axis shows the spectral radius, the y-axis the phase change in every connection. The darker the colour, the better the performance.

4. CONCLUSIONS

In this paper we have investigated a network of coupled SOAs as a reservoir for RC by means of evaluating this kind of reservoir on a benchmark speech recognition task. It turns out that SOA reservoirs can be used to solve such kind of complex problems, despite the limitations imposed by a practical implementation. In the future, a practical demonstration of a chip of SOAs used as a photonic reservoir will be further pursued.

ACKNOWLEDGEMENTS

K. Vandoorne acknowledges the Special Research Fund (BOF) of Ghent University for a specialization grant. This work has been carried out in the framework of the IAP project Photonics@be of the Belgian Science Policy.

REFERENCES

[1] H. Jaeger, H. Haas: Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, *Science*, vol. 304, pp. 78-80, Apr. 2 2004.

[2] W. Maass, T. Natschlager, H. Markram: Real-time computing without stable states: A new framework for neural computation based on perturbations, *Neural Computation*, vol. 14, pp. 2531-2560, Nov. 2002.

[3] R. Lyon: A computational model of filtering, detection and compression in the cochlea, in *Proc. IEEE ICASSP*, Paris, May 1982, pp. 1282-1285.

[4] D. Verstraeten, PhD thesis: *Reservoir Computing: Computation with Dynamical Systems*, in PARIS, Dept. of Electronics and Information Systems Ghent: Ghent University, 2009.

[5] G. P. Agrawal, N. A. Olsson: Self-phase modulation and spectral broadening of optical pulses in semiconductor-laser amplifiers, *IEEE Journal of Quantum Electronics*, vol. 25, pp. 2297-2306, Nov 1989.