



ÉCOLE
POLYTECHNIQUE
DE BRUXELLES



UNIVERSITÉ LIBRE DE BRUXELLES

Interferometric stabilisation of a fibre-based optical computer

Experimental study

Mémoire présenté en vue de l'obtention du diplôme
d'Ingénieur Civil physicien à finalité spécialisée

Denis Verstraeten

Directeur

Professeur Marc Haelterman

Co-Promoteur

Professeur Serge Massar

Superviseur

Lorenz Butschek

Service

Opera

Année académique
2018 - 2019

Abstract

Acknowledgements

Contents

1	Introduction	5
2	Reservoir Computing	6
2.1	Introduction	6
3	Optical RC with frequency multiplexed neurons	9
4	Interferometric stabilisation of RC optical resonator	10
5	Results	11
6	Conclusion	12
	Acronyms	13

Chapter 1

Introduction

For the past few years, interest in optical data processing devices has been increasing. Their main advantage over silicon-based computers is that they are intrinsically faster because the information is carried around at nearly the speed of light, which could allow to overcome the limit in processing speed soon to be reached by classical integrated circuit electronics.

This Master thesis tackles the implementation of an optical computer based on reservoir computing.

Chapter 2

Reservoir Computing

2.1 Introduction

Reservoir Computing (RC) is a bio-inspired artificial Recurrent Neural Network (RNN) which is based on the Echo State Network (ESN) introduced by Herbert Jaeger in [4]. This computation scheme is well suited for real-time data processing and for chaotic time series prediction[4, 5, 7], and achieves state of the art performances in those domains, as well as in speech recognition[10, 9, 6], nonlinear channel equalisation[4] and financial forecasting [1].

A Reservoir Computer (RC) is specific kind of Neural Network (NN), which is a computation paradigm mimicking the behaviour of a biological brain. As can be seen on Figure 2.1a, the artificial neurons are merely interconnected entities carrying an activation level. As shown on Figure 2.1b, the activation level of a neuron is updated according to the connection weights of the network, also known as the synaptic matrix, and with a nonlinear function, called the *activation function*, which allows NN to perform classification tasks [2, p.225][8, p.727].

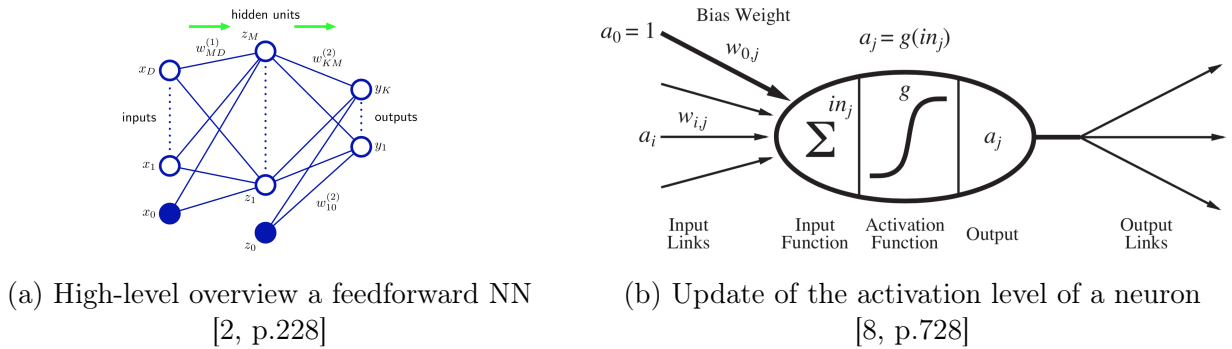


Figure 2.1: Schematic representation of a Neural Network.

The activation level of the neurons making up the reservoir characterise its state, which is a time-dependent object. The neurons are interconnected in such a way that they influence the dynamic of each other, leading to a complicated evolution of the state of the reservoir. What a first glance may seem to be a mathematical nightmare turns out to be the main advantage of RC. Indeed, by making the connection matrix as messy as possible, *i.e.* by using randomness, breaking symmetries,... one notices that the effect of such a reservoir, when being fed a time-dependent signal into the input neurons, is to map it to a higher-dimensional dynamics space. RC reach their best performance when they are used in the echo state, which is a regime where the transients caused by the inputs are neither amplified nor damped, somehow providing a

memory to the reservoir [3]. The output of a RC is obtained by adequately combining the activation state of each neurons. The ideas developed in this paragraph are illustrated on Figure 2.2.

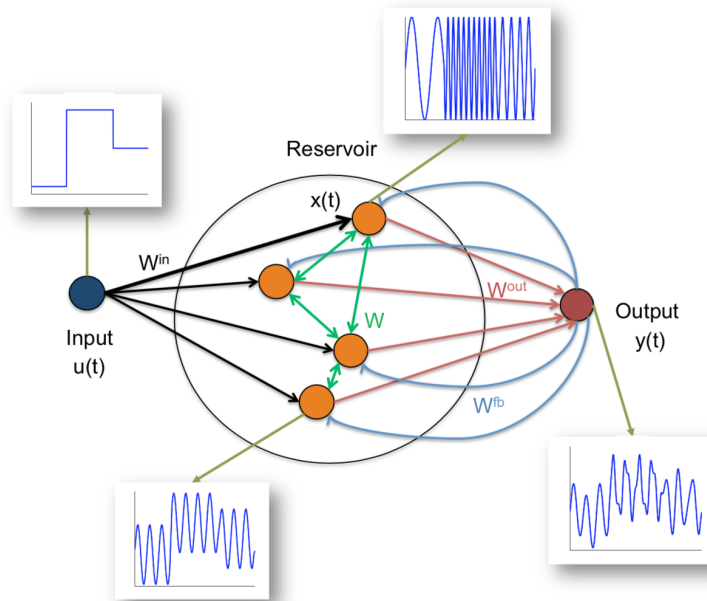
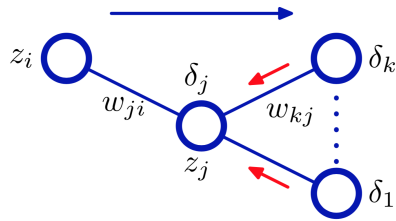
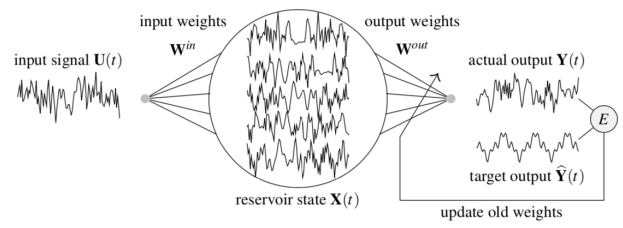


Figure 2.2: Principle scheme of a Reservoir Computer [1]

Regardless of the learning scheme used to train a NN, the basic idea is always to minimise the difference between the desired and the actual outputs. In practice, this is achieved by updating the coefficients in the synaptic matrix of the NN. On Figure 2.3a, the learning procedure for a small portion of a feed-forward NN, for instance the one from Figure 2.1a, is shown. The NN is fed with data from the left, as can be seen with the blue arrows. The red arrows represent the error on the output being *backpropagated*, which means that the learning algorithm evaluates how each connection weight of the synaptic matrix should be updated in order to decrease the output error [2, p.241]. This procedure often turns out to be a really complicated task, which explains why the development of efficient Machine Learning (ML) algorithms is such a hot topic nowadays. On the other hand, as can be seen on Figure 2.3b, RC only need their output weights to be adjusted when being trained, which makes them computationally lighter. This is due to the fact the connections of the reservoir should not contain any information about the task, but should only be used to reach the ESN regime, as previously mentioned. There are two main families of training methods for RC. On the one hand, there is the *batch learning*, which comprises the methods requiring to store a bunch of data regarding the task being taught before being able to actually compute the output weights. Once enough data is gathered, this kind of algorithms returns the optimal weights all at once. They present the advantage of involving only one training phase, after which the RC are ready to perform. However, the need for vast amount of data and the inability for the RC to adapt to an input evolving out of the range for which it has been trained are two drawbacks. On the other hand, *online learning* methods allow to iteratively improve the output weights. Therefore, starting from a first guess, these algorithms will converge to workable output weights. They are much more adaptable than the batch learning ones, however, their convergence is not guaranteed and can be slow.



(a) Neural Network [2, p.244]



(b) Reservoir Computer [3]

Figure 2.3: Machine Learning for different kinds of Neural Networks

Chapter 3

Optical RC with frequency multiplexed neurons

Chapter 4

Interferometric stabilisation of RC optical resonator

Chapter 5

Results

Chapter 6

Conclusion

Acronyms

ESN Echo State Network 6, 7

ML Machine Learning 7, 8

NN Neural Network 6–8

RC Reservoir Computer 4, 6–10

RC Reservoir Computing 4, 6

RNN Recurrent Neural Network 6

Bibliography

- [1] A. Bernal, S. Fok, and R. Pidaparthi. “Financial Market Time Series Prediction with Recurrent Neural Networks”. In: (2012). URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.278.3606&rep=rep1&type=pdf>.
- [2] Christopher Bishop. *Pattern recognition and machine learning*. New York: Springer, 2006. ISBN: 978-0387-31073-2.
- [3] Alireza Goudarzi et al. “A Comparative Study of Reservoir Computing for Temporal Signal Processing”. In: *CoRR* abs/1401.2224 (2014).
- [4] H. Jaeger. “Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication”. In: *Science* 304.5667 (Apr. 2004), pp. 78–80. DOI: 10.1126/science.1091277. URL: <https://doi.org/10.1126/science.1091277>.
- [5] H. Jaeger. *The "echo state" approach to analysing and training recurrent neural networks*. 2001.
- [6] Herbert Jaeger et al. “Optimization and applications of echo state networks with leaky-integrator neurons”. In: *Neural Networks* 20.3 (Apr. 2007), pp. 335–352. DOI: 10.1016/j.neunet.2007.04.016. URL: <https://doi.org/10.1016/j.neunet.2007.04.016>.
- [7] M. Lukoševičius, M. Jaeger, and B. Schrauwen. “Reservoir Computing Trends”. In: *KI - Künstliche Intelligenz* 26.4 (May 2012), pp. 365–371. DOI: 10.1007/s13218-012-0204-5. URL: <https://doi.org/10.1007/s13218-012-0204-5>.
- [8] Stuart Russell. *Artificial intelligence : a modern approach*. Upper Saddle River, New Jersey: Prentice Hall, 2010. ISBN: 978-0-13-604259-4.
- [9] Fabian Triefenbach et al. “Phoneme Recognition with Large Hierarchical Reservoirs”. In: *Advances in Neural Information Processing Systems 23*. Ed. by J. D. Lafferty et al. Curran Associates, Inc., 2010, pp. 2307–2315. URL: <http://papers.nips.cc/paper/4056-phoneme-recognition-with-large-hierarchical-reservoirs.pdf>.
- [10] D. Verstraeten, B. Schrauwen, and D. Stroobandt. “Reservoir-based techniques for speech recognition”. In: *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. IEEE, 2006. DOI: 10.1109/ijcnn.2006.246804. URL: <https://doi.org/10.1109/ijcnn.2006.246804>.