Neural Networks 55 (2014) 59–71

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Stochastic nonlinear time series forecasting using time-delay reservoir computers: Performance and universality

Lyudmila Grigoryeva [a], Julie Henriques [b], Laurent Larger [c], Juan-Pablo Ortega [d,*]

[a] Laboratoire de Mathématiques de Besançon, Université de Franche-Comté, UFR des Sciences et Techniques. 16, route de Gray. F-25030 Besançon Cedex, France

[b] Cegos Deployment. 11 rue Denis Papin. F-25000 Besançon, France

[c] UMR CNRS FEMTO-ST 6174/Optics Department, Université de Franche-Comté, UFR des Sciences et Techniques. 16, route de Gray. F-25030 Besançon Cedex, France

[d] Centre National de la Recherche Scientifique, Laboratoire de Mathématiques de Besançon, Université de Franche-Comté, UFR des Sciences et Techniques. 16, route de Gray. F-25030 Besançon Cedex, France

ARTICLE INFO

Article history:
Received 19 November 2013
Received in revised form 11 February 2014
Accepted 13 March 2014
Available online 21 March 2014

Keywords:
Reservoir computing
Echo state networks
Neural computing
Time-delay reservoir
Time series forecasting
Universality
VEC-GARCH model
Volatility forecasting
Realized volatility
Parallel reservoir computing

ABSTRACT

Reservoir computing is a recently introduced machine learning paradigm that has already shown excellent performances in the processing of empirical data. We study a particular kind of reservoir computers called time-delay reservoirs that are constructed out of the sampling of the solution of a time-delay differential equation and show their good performance in the forecasting of the conditional covariances associated to multivariate discrete-time nonlinear stochastic processes of VEC-GARCH type as well as in the prediction of factual daily market realized volatilities computed with intraday quotes, using as training input daily log-return series of moderate size. We tackle some problems associated to the lack of task-universality for individually operating reservoirs and propose a solution based on the use of parallel arrays of time-delay reservoirs.

1. Introduction

Reservoir computing (RC), also referred to in the literature as liquid state machines, echo state networks, or nonlinear transient computing, is a recently introduced machine learning paradigm (Crook, 2007; Jaeger, 2001; Jaeger & Haas, 2004; Lukoševičius & Jaeger, 2009; Maass, 2011; Maass, Natschläger, & Markram, 2002; Verstraeten, Schrauwen, D'Haene, & Stroobandt, 2007) that has already shown excellent performances in the processing of empirical data. Examples of tasks that have already been explored are spoken digit recognition (Appeltant et al., 2011; Brunner, Soriano, Mirasso, & Fischer, 2013; Jaeger, Lukoševičius, Popovici, & Siewert, 2007; Larger et al., 2012; Paquot et al., 2012), the NARMA model identification task (Atiya & Parlos, 2000; Rodan & Tino, 2011), and continuation of chaotic time series (Jaeger & Haas, 2004).

RC is a promising and fundamentally new approach to neural computing that can be seen as a modification of the traditional artificial recurrent neural networks (RNN) in which the architecture and the neuron weights of the network are created using different static or dynamic procedures and remain inaccessible during the training stage; this out of reach RNN is called the reservoir. The output signal is obtained out of the RNN via a linear readout layer that is trained using the teaching signal by means of a ridge regression.

In this paper we concentrate on a specific class of reservoirs whose architecture is dynamically created via the sampling of the solutions of a forced time-delay differential equation of the form

$$\dot{x}(t) = F(x(t), x(t-\tau), I(t)), \tag{1.1}$$

where $\tau > 0$ is the delay, and the external forcing $I(t)$ is used to insert the input signal that needs to be processed into the reservoir. We will refer to reservoirs constructed this way as time-delay reservoirs (TDRs). As we will see in detail in the next section, this procedure yields reservoir topologies that generalize the so

* Corresponding author. Tel.: +33 670626326.
E-mail addresses: Lyudmyla.Grygoryeva@univ-fcomte.fr (L. Grigoryeva), jhenriques@deployment.org (J. Henriques), Laurent.Larger@univ-fcomte.fr (L. Larger), Juan-Pablo.Ortega@univ-fcomte.fr (J.-P. Ortega).

http://dx.doi.org/10.1016/j.neunet.2014.03.004
0893-6080/

called Simple Cycle Reservoirs (SCR) (Gutiérrez, San-Martín, Ortín, & Pesquera, 2012; Rodan & Tino, 2011) that have already shown good performances in a variety of tasks. Moreover, as equations of the type (1.1) appear profusely in the modeling of concrete phenomena, this leads to the possibility of building physical realizations of the reservoir (Appeltant et al., 2011; Brunner et al., 2013; Dambre, Verstraeten, Schrauwen, & Massar, 2012; Duport, Schneider, Smerieri, Haelterman, & Massar, 2012; Larger et al., 2012; Paquot et al., 2012) which confers interesting possibilities to this approach. A particularly impressive example related to this feature are the results presented in Brunner et al. (2013) where an optoelectronic implementation of a TDR is capable of achieving the lowest documented error in the speech recognition task as well as the highest classification speed in an experiment design in which the setup carries out in parallel digit and speaker recognition.

Our work contains three main contributions:

- We show the pertinence of using TDRs in the forecasting of multivariate discrete time nonlinear stochastic processes, which is a non-deterministic extension of the frameworks where this technique has already proved to exhibit good performance. We carry out this assessment by exploring the applicability of this approach in volatility forecasting both with synthetic data generated using VEC-GARCH type models, as well as with actual market daily realized volatility matrices computed using intraday market price quotes. RC has already been used in the forecasting of factual time series from various origins (see for example Ilies et al. (2007), Wyffels, Schrauwen, and Stroobandt (2008), Wyffels and Schrauwen (2010) and references therein); the underlying dynamics of the series chosen in most of these works exhibits significant seasonal and trend components that help in the forecasting. In contrast with these existing works, we will be dealing in our work with nonlinearly generated multidimensional white noise that lacks those helping features.
- We evidence the lack of task-universality for an individually operating TDR; more specifically, a reservoir that performs well in the forecasting of a stochastic dynamical model for a given horizon does not reproduce the same performance when the parameter values are modified and, to a lesser extent, when the prediction horizon changes. This observation has two serious negative consequences:
  - Adapting the reservoir to different tasks, even when they are closely related (like the forecasting of the same dynamical phenomenon at different horizons), requires the execution of expensive cross-validation based optimization routines.
  - It prevents the use of TDRs in a multi-tasking regime, which is one of the major potential advantages of reservoir computing (Maass, 2011).
- We propose the use of parallel arrays of reservoir computers in order to limit the computational effort necessary to adapt the device to the task at hand, to improve the performance for small training sample sizes, and to achieve robustness with respect to modifications in the task, hence making it more universal and appropriate for parallel reading. This approach is inspired by the theoretical results in Boyd and Chua (1985), Maass (2011), Maass and Sontag (2000) that use basis filters to secure the defining features of a functioning RC, namely, the fading memory, separation, and approximation properties. The use of several reservoirs, each running at different regimes, functionally enriches the output of the global device. A parallel approach similar to ours with only two reservoirs has been explored in Ortin, Pesquera, and Gutiérrez (2012) where it is shown that the use of parallel nodes with different parameters is capable of providing memory capacity increases.

The paper is structured as follows: Section 2 explains in detail the construction of TDR computers and how they are adapted for forecasting tasks. Section 3 presents the volatility forecasting task that we are interested in, introduces the parametric family that we will be using as data generating process, and reviews the standard forecasting approach in this context that we use as a performance benchmark; additionally we introduce the parallel reservoir architecture that helps us solve challenges having to do with universality issues and small training sizes. The empirical results are presented in detail in Section 4. Section 5 concludes the paper.

2. Construction of time-delay reservoir (TDR) computers

A reservoir computer is formed by three modules that are depicted in Fig. 1: an input layer (module A in the figure) that feeds the signal to be treated, the reservoir (module B) that processes it, and a readout layer (module C) that uses the reservoir output to produce the information needed. The input and readout layers are adapted to the specific features of the task under consideration; for example the construction of the input layer needs to take into consideration the dimensionality of the signal that needs to be processed; different ways to feed the signal can have an impact on the training speed of the system. Regarding the readout layer, we restrict to linear readings that are calibrated by minimizing the mean square error committed when comparing the output with a teaching signal, via a ridge regression. In this work we consider exclusively an offline (batch) type of training.

Construction of the time-delay reservoir (module B in Fig. 1). We start by describing the construction of the dynamical time-delay reservoir out of a solution $x(t)$ of the differential equation (1.1). The state of the reservoir is labeled in time steps that are multiples of the delay τ and is characterized by the values of N *virtual neurons* obtained out of the regular sampling of $x(t)$ during a given time-delay interval. We will denote by $x_i(k)$ the *value* of the ith neuron of the reservoir at time $k\tau$ defined by

$$x_i(k) := x(k\tau - (N-i)\theta), \quad i \in \{1, \dots, N\}, \tag{2.1}$$

where $\tau := \theta N$ and θ is referred to as the *separation between neurons*. We will also say sometimes that $x_i(k)$ is the *ith neuron value of the kth layer of the reservoir*.

Even though the justification for the TDR topology depicted in Fig. 1 intuitively follows from the form of the differential equation (1.1), it can be also explained through its Euler time discretization as in Appeltant et al. (2011), Gutiérrez et al. (2012). More explicitly, consider the following special case of the time-delay differential equation (1.1):

$$\dot{x}(t) = -x(t) + f(x(t-\tau), I(t)). \tag{2.2}$$

Given an arbitrary natural number N, the Euler discretization of (2.2) with integration step $\theta := \tau/N$ is

$$\frac{x(t) - x(t-\theta)}{\theta} = -x(t) + f(x(t-\tau), I(t)). \tag{2.3}$$

We now construct a reservoir by using the same assignment of neuron values as in (2.1), that is, we define

$$x_i(k) := x(k\tau - (N-i)\theta) \qquad \text{and} \qquad I_i(k) := I(k\tau - (N-i)\theta).$$

The Eq. (2.3) amounts to saying that the neuron values $x_i(k)$ satisfy the following recursive relation:

$$x_i(k) := e^{-\xi} x_{i-1}(k) + (1 - e^{-\xi}) f(x_i(k-1), I_i(k)), \tag{2.4}$$

with $x_0(k) := x_N(k-1)$, and $\xi := \log(1+\theta)$, that shows how, as depicted in Fig. 1, the ith neuron value of the kth layer of the reservoir is a convex linear combination between the previous