Procedia Computer Science

Volume 41, 2014, Pages 176–181

This space is reserved for the Procedia header, do not use it

BICA 2014. 5th Annual International Conference on Biologically Inspired Cognitive Architectures

Towards a Calculus of Echo State Networks

Alireza Goudarzi[1] and Darko Stefanovic[1]

[1] Department of Computer Science, University of New Mexico, Albuquerque, New Mexico, USA
alirezag@cs.unm.edu

[2] Center for Biomedical Engineering, University of New Mexico, Albuquerque, New Mexico, USA
darko@cs.unm.edu

Abstract

Reservoir computing is a recent trend in neural networks which uses dynamical perturbations in the phase space of a system to compute a desired target function. We show one can formulate an expectation of system performance in a simple model of reservoir computing called echo state networks. In contrast with previous theoretical frameworks, which uses annealed approximation, we calculate the exact optimal output weights as a function of the structure of the system and the properties of the input and the target function. Our calculation agrees with numerical simulations. To the best of our knowledge this work presents the first exact analytical solution to optimal output weights in echo state networks.

Keywords: Reservoir computing, echo state networks, analytical training, Wiener filters, dynamics

1 Introduction

In this paper we report our preliminary results in building a framework for a mathematical study of reservoir computing (RC) architecture called the echo state network (ESN). Reservoir computing (RC) is a recent approach in time series analysis that uses the transient dynamics of a system, as opposed to its stable states, to compute a desired function. The classic example of reservoir computing is the echo state network, a recurrent neural network with random structure. These networks have shown good performance in many signal processing applications. The theory of echo state networks treats memory capacity [4], (how long can the network remember its inputs), and the echo state property [16], (long term convergence of the phase space of the network). In RC, computation relies on the dynamics of the system and not its specific structure, which makes the approach an intriguing paradigm for computing with unconventional and neuromorphic architectures [11–13]. In this context, our vision is to develop special-purpose computing devices that can be trained or "programmed" to perform a specific task. Consequently, we would like to know the expected performance of a device with a given structure on given a task. Echo state networks give us a simple model to study reservoir computing. Extant studies of computational capacity and performance of ESN for

Selection and peer-review under responsibility of the Scientific Programme Committee of BICA 2014

© The Authors. Published by Elsevier B.V.
doi:10.1016/j.procs.2014.11.101

various tasks have been carried out computationally and the main theoretical insight has been the upper bound for linear memory capacity [5,10,15].

Our aim is to develop a theoretical framework that allows us to form an expectation about the performance of RC for a desired computation. To demonstrate the power of this framework, we use it to calculate the optimal weights for ESN to reconstruct its previous inputs. Whereas previous attempts used the annealed approximation method to simplify the problem [15], we derive an exact solution for the optimal output weights for a given system. Our formulation reveals that ESN computes the output as a linear combination of the correlation structure of the corresponding input signal and therefore the performance of ESN on a given task will depend on how well the output can be described as the input correlation in various time scales. Full development of the framework will allow us to extend our predictions to more complex tasks and more general RC architectures.

2 Background

In RC, a high-dimensional dynamical core called a *reservoir* is perturbed with an external input. The reservoir states are then linearly combined to create the output. The readout parameters are calculated by regression on the state of a teacher-driven reservoir and the expected output. Unlike other forms of neural computation, computation in RC takes place within the transient dynamics of the reservoir. The computational power of the reservoir is attributed to a short-term memory created by the reservoir [8] and the ability to preserve the temporal information from distinct signals over time [9]. Several studies attributed this property to the dynamical regime of the reservoir and showed it to be optimal when the system operates in the critical dynamical regime—a regime in which perturbations to the system's trajectory in its phase space neither spread nor die out [1–3,14]. The reason for this observation remains unknown. Maass et al. [9] proved that given the two properties of *separation* and *approximation*, a reservoir system is capable of approximating any time series. The separation property ensures that the reservoir perturbations from distinct signals remain distinguishable, whereas the approximation property ensures that the output layer can approximate any function of the reservoir states to an arbitrary degree of accuracy. Jaeger [7] proposed that an ideal reservoir needs to have the so-called echo state property (ESP), which means that the reservoir states asymptotically depend on the input and not the initial state of the reservoir. It has also been suggested that the reservoir dynamics acts like a spatiotemporal kernel, projecting the input signal onto a high-dimensional feature space [6].

3 Model

Here we restrict attention to linear ESNs, in which both the transfer function of the reservoir nodes and the output layer are linear functions, Figure 1. The readout layer is a linear combination of the reservoir states. The readout weights are determined using supervised learning techniques, where the network is driven by a teacher input and its output is compared with a corresponding teacher output to estimate the error. Then, the weights can be calculated using any closed-form regression technique [10] in offline training contexts. Mathematically, the input-driven reservoir is defined as follows. Let N be the size of the reservoir. We represent the time-dependent inputs as a column vector $\mathbf{u}(t)$, the reservoir state as a column vector $\mathbf{x}(t)$, and the output as a column vector $\mathbf{y}(t)$. The input connectivity is represented by the matrix $\mathbf{V}$ and the reservoir connectivity is represented by an $N \times N$ weight matrix $\mathbf{W}$. For simplicity, we

Download English Version:

https://daneshyari.com/en/article/486322

Download Persian Version:

https://daneshyari.com/article/486322

Daneshyari.com