Information Sciences 364–365 (2016) 184–196

Contents lists available at ScienceDirect

Information Sciences

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

Modeling neural plasticity in echo state networks for classification and regression

Mohd-Hanif Yusoff, Joseph Chrol-Cannon, Yaochu Jin*

Department of Computer Science, University of Surrey, Guildford GU2 7XH Surrey, UK

A R T I C L E I N F O

Article history:
Received 9 July 2015
Revised 31 October 2015
Accepted 9 November 2015
Available online 21 November 2015

Keywords:
Echo state networks
Synaptic plasticity
Learning algorithms
Online learning
Offline learning

A B S T R A C T

Echo state networks (ESNs) are one of two major neural network models belonging to the reservoir computing framework. Traditionally, only the weights connecting to the output neuron, termed read-out weights, are trained using a supervised learning algorithm, while the weights inside the reservoir of the ESN are randomly determined and remain unchanged during the training. In this paper, we investigate the influence of neural plasticity applied to the weights inside the reservoir on the learning performance of the ESN. We examine the influence of two plasticity rules, anti-Oja's learning rule and the Bienenstock–Cooper–Munro (BCM) learning rule on the prediction and classification performance when either offline or online supervised learning algorithms are employed for training the read-out connections. Empirical studies are conducted on two widely used classification tasks and two time series prediction problems. Our experimental results demonstrate that neural plasticity can more effectively enhance the learning performance when offline learning is applied. The results also indicate that the BCM rule outperforms the anti-Oja rule in improving the learning performance of the ENS in the offline learning mode.

1. Introduction

In comparison to feed forward neural networks (FNNs), recurrent neural networks (RNNs) are known to have richer dynamics [19]. For a given input signal, an RNN is able to produce a nonlinear transformation of the input history within its internal state via the recurrent connection pathways. With such memory of dynamics, RNNs are able to offer some advantages over FNNs for temporal information processing. Despite the attractive rich dynamics of RNNs, effective training of RNNs remains challenging [3].

ESNs introduce a new paradigm in the training of RNNs for solving supervised learning problems [19]. An ESN can be seen as a special case of multi-layer neural network, which consists of an input layer, a hidden layer and an output layer. The hidden layer, commonly known as the internal reservoir, consists of a set of units forming a recurrent neural network with sparse connectivity. The output layer is composed of a unit known as the readout neuron. One unique property of the ESN lies in the way in which the network updates its weights, i.e., only the readout connections are updated, whilst the internal weights of the reservoir are fixed [18]. Typically, a linear regression algorithm is applied to train the readout weights in ESNs [11]. The randomized structure and linear training rule, taken together, make ESNs a very efficient learning model.

The advantage of efficient learning has made ESNs very effective for solving different kinds of task such as pattern classification, e.g. [15,26] and time-series prediction, e.g. [25,33], in addition to many others. However, it has also been found nontrivial

* Corresponding author. Tel.: +441483686037.
E-mail address: yaochu.jin@surrey.ac.uk (Y. Jin).

http://dx.doi.org/10.1016/j.ins.2015.11.017

to further improve the learning performance of ESNs [3]. One potential solution to this issue is to modulate the neural dynamics of the reservoir by tuning the randomly determined weights.

Recently, neural models based on random projections of the input features have become a popular way of simplifying the training of nonlinear models applied to pattern recognition and regression tasks [1,4]. In addition, biologically inspired neural models [9,16] are becoming more widely used for increasing the adaptation of randomized learning models to the input data. In this work, the reservoir computing paradigm is used to combine the features of unsupervised biological adaptation with the single-layer training approach.

Computational modeling of neural plasticity and its role in self-organization of artificial neural network models have been widely investigated [6,7]. However, not much research has been reported on comparing the influence of different kinds of neural plasticity for optimizing the reservoir connections. Studies have typically focused on using intrinsic plasticity [29,30], to improve learning performance of reservoir based neural network models, with only a few exceptions that study other forms of reservoir adaptation [2,3]. In this paper, we investigate the interaction between the plasticity in an ESN reservoir and the adaptation of its readout connections. Two plasticity rules, the anti-Oja rule [3] and BCM rule [5] are implemented in the ESN for unsupervised learning of the internal weights inside the reservoir. The novelty here is that we are empirically comparing Hebbian and anti-Hebbian forms of biologically inspired plasticity. Also, as far as we are aware, this is the first application of the BCM learning rule, to an ESN model. Once the unsupervised learning of the weights inside the reservoir is complete, a supervised learning algorithm is applied to the read-out weights. Here, two supervised learning approaches have been investigated, on-line and off-line. By combining one of the two plasticity rules with one of the two supervised learning approaches, four different learning scenarios in total have been studied on two classification tasks, breast cancer diagnosis and adult census income, and two time series prediction problems, the sunspot time series and the Mackey-Glass time series. The empirical comparison between online and offline learning modes is another area of novelty for this work, as we show that online learning is particularly unstable when the ESN connection matrix has a high spectral radius. Our results also demonstrate that BCM adaptation increases the spectral radius, making it a poor choice for use with an un-regularized online learning model. Our results indicate that application of either of the plasticity rules is able to enhance the subsequent supervised learning of the readout connections, provided that the supervised learning is conducted off-line.

2. Echo state networks

2.1. The ESN architecture

Consider a network consisting of K input neurons, N hidden (i.e. reservoir) neurons and L readout neurons. The ESN architecture is illustrated in Fig. 1. Synaptic connections are separated into three types: input-to-reservoir (W^{in}), internal recurrent (W^{res}), and reservoir-to-readout (W^{out}). In the basic architecture of ESN, there can be optional W^{ofb} and W^{in} that directly connect to the readout neurons. However, in this study we do not implement these feedback or direct input-to-output connections.

In the ESN, the activation states of the reservoir units, x, are updated using

$$x(t+1) = f(W^{in}u(t+1) + W^{res}x(t)) \tag{1}$$

where t is a time step of the learning sample and f is the neuron activation function of the reservoir unit. In the experiments reported in this paper, f is a tansigmoid.

The readout, y is computed using (2):

$$y(t+1) = W^{out}(u(t+1), x(t+1), y(t)) \tag{2}$$

where $u(t+1), x(t+1), y(t)$ is the concatenation of the input, internal (reservoir), and previous output activation vectors [11].

2.2. Training the readout neuron

In this section, we consider both offline [11] and online learning [24] for training the readout weights. The plasticity rules are only used to adapt the weights of connections inside the reservoir. The readout weights are trained separately using a supervised learning algorithm after the unsupervised learning of the weights inside the reservoir is complete.

In the offline learning mode, the readout weights are updated using all of the training data. In this study, we used the least square estimation (LSE) method [8] to calculate the optimal weight values for the readout connections that minimize the difference between the desired (target) output signal and the actual output of the reservoir neurons in response to the input vectors. The training works in a single step computation to minimize the following:

$$E(W^{out}) = 1/T \sum_{t=1}^{T} ||(y^{\mathrm{desired}}(t) - -y(t))||^2, \tag{3}$$

where E is the error on the training data for the given weights W^{out}, y^{desired} is the desired output, and y is the actual output of the network.

The LSE can be described by;

$$W^{out} = (RR^{\mathrm{T}})^{-1}R \cdot y^{\mathrm{desired}}(t) \tag{4}$$