Information Sciences 000 (2015) 1–13

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

ARTICLE INFO

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

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

ABSTRACT

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
0020-0255/

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:

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

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;

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

Please cite this article as: M.-H. Yusoff et al., Modeling neural plasticity in echo state networks for classification and regression, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.11.017

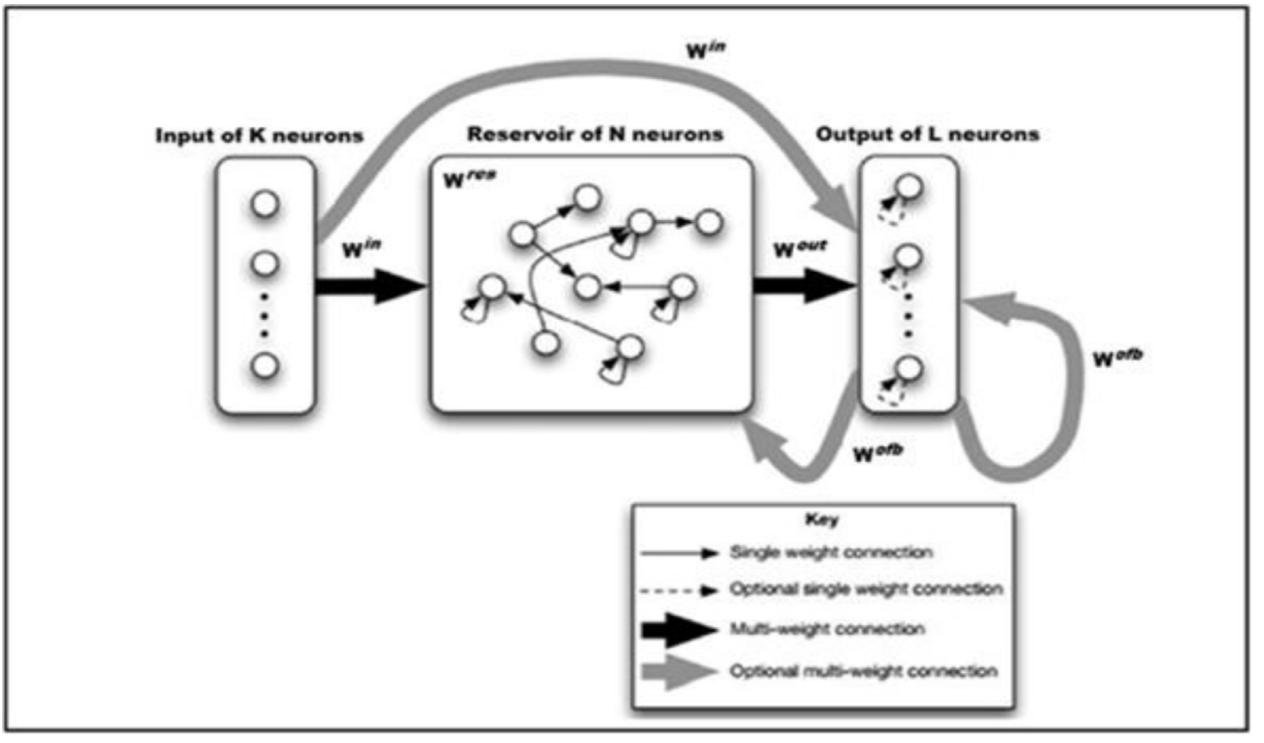

Fig. 1. The basic architecture of echo state network model adapted from [11].

where R is a matrix describing the reservoir firing activity over time, which is the state of x at time $t = 1, 2, \ldots, T$ multiplied with $y^{\text{desired}}(t)$, a vector denoting the desired output of the read-out neuron over time $t = 1, 2, \ldots, T$.

The LSE is numerically stable method for training the readout weights by minimizing the mean squared error between the output predicted by the network and the desired output of all training data. This type of learning algorithms is well known as a batch learning method [8].

In the online learning mode, the readout weights are trained by minimizing the error between the desired output and the real output when the training samples are presented sequentially. The delta rule is used for online learning, which is a stochastic gradient descent method originally used for updating the weights of a single-layer perceptron model [24]. The delta rule is described as follows:

$$\Delta W = \eta(y^{\text{desired}}(t) - -y(t))(x(t)) \tag{5}$$

$$W^{out} = W^{out} + \Delta W \tag{6}$$

where η is the learning rate and t is the time step of the learning iterations, $t = 1, 2, \ldots, T$. $x(t)$ is the vector of neuron firing activation states of x at time step t. This mechanism computes the incremental adaptation of readout weights.

2.3. Parameters in ESNs

In this section, we highlight the important factors of reservoir dynamics which play a major role in constructing the ESN model. We focus on the most important parameters that affect the reservoir network and the individual neurons. The aim of the study on parameter settings is to understand the influence of these parameters on the whole learning process of ESNs for classification and time series prediction.

- The size of reservoir

Commonly, the selection of the size of reservoir, i.e., the number of internal neurons, is the most challenging in generating a suitable ESN model. That is because the size (N) of the reservoir determines the numbers of connection weights (N^2) in the reservoir network. Furthermore, the reservoir size represents the determined memory capacity of the network model [14], whereas the dynamic memory properties are determined by the size of spectral radius. Many findings in the literature suggest that having a large number of reservoir neurons can offer advantages in solving challenging problems [12]. Moreover, having randomly connected neurons enhances the projection of nonlinear transformations from input signals that can be used to form the output signal. However, an ESN with an extremely high number of internal neurons may suffer from over-fitting. The optimal value for N is a function of task complexity and the size of the available training data [12].

Please cite this article as: M.-H. Yusoff et al., Modeling neural plasticity in echo state networks for classification and regression, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.11.017

Note that an advantage of the ESN is that the training complexity is only linearly dependent on the number of the internal neurons for the case of online iterative learning, while such dependence is quadratic in the case of classic recurrent neural networks trained with gradient decent methods [13].

- The connection density and spectral radius

The most important feature of a "good" reservoir is stable and rich neuronal dynamics. To guarantee the stability of the activity dynamics, we must control the size of the spectral radius α, which is the largest absolute eigenvalue of the connection weight matrix, W^{res} that connects all the individual reservoir neurons [31]. Before creating the reservoir network, we have to make sure that the spectral radius, α, must be smaller than unity, i.e., $\alpha < 1$, to ensure the stability of the network and the occurrence of the echo state property [11].

Another parameter, connectivity ratio (c) of the reservoir, is also very important for the stability of the reservoir, because when we scale the reservoir weights, W^{res}, we have to ensure that α remains at a desired level which is lower than 1. As a result, for a given value of α, the network will be either densely connected with smaller connecting weights, or sparsely connected with higher connecting weights [28]. Stability can be maintained in either case, however, other characteristics of the network may change, and an excessive connectivity ratio will lead to stronger coupling of neural internal states and reduce the diversity of the neuronal states in the reservoir. It is common to hold the connectivity ratio at a constant and low level, usually between 0.01 and 0.2, while the spectral radius is optimized for a given task, a value usually between 0.5 and 1.0. In that way, the richness of the internal nonlinear states is ensured by sparse connectivity, while the optimal fading memory effect is determined by finding the appropriate spectral radius [17].

- Input scaling

In addition to the reservoir size and the connectivity ratio, we need to consider scaling the input and feedback signals. This is also a crucial part in developing a reservoir network, which considerably affects the learning performance [17]. This is due to the fact that scaling of the input signal will determine whether the system works in the linear mode (input will use only the linear region of the sigmoid function), binary mode (large input will drive the neural outputs to extreme values of the sigmoid function $\{-1; 1\}$), or nonlinear mode (optimally scaled input uses the entire curvature of the sigmoid activation function). The latter mode is generally desired when modeling chaotic systems.

Both global and local input scaling are critical, which are both regulated by the input weights vector and feedback weights vector, i.e. W^{in} and W^{back}. Since these vectors are initialized randomly within the range $\{-1;1\}$, a proper global scaling should ensure that the reservoir contains rich combinations of neurons ranging from linear, through nonlinear and to binary ones [12].

Nevertheless, this scaling aspect can be further optimized according to input dimension in order to produce optimal performance. In case of a high dimensional multivariate input, a fine tuning of the scaling method may be necessary. Additionally, modification of the input scaling and the spectral radius can determine whether the reservoir network operates in an input-driven or intrinsic-state driven mode.

Low-volume external input combined with a large spectral radius will excite the neurons regulated frequently by the inputs projected from the nearest reservoir neurons, and hence the whole network will be less sensitive and more autistic. On the other hand, high-volume external input and a large spectral radius will increase the risk of instability [22].

3. Plasticity rules

The main goal of this study is to examine the effectiveness of applying synaptic plasticity to the internal weights of the ESN in order to enhance the learning performance of ESNs for classification or time-series prediction. In the standard ESN model defined in [11], the supervised learning process for the readout weights is performed without applying plasticity to the reservoir. The connectivity of the reservoir is randomly generated, along with the connecting weights within the reservoir. However, we hypothesize that the optimal reservoir connectivity depends on the problem to be solved. A plasticity mechanism is able to modify the strength of the synapses within the reservoir based on the activities stimulated by the input. By doing this, it is hoped that the structural information embedded in the input signal can be learned with the help of neural plasticity. To investigate the influence of plasticity rules on the learning performance of ESNs, we examine two different synaptic plasticity rules that are expected to improve the prediction performance, the anti-Oja learning rule [3] and the Bienenstock–Cooper–Munro (BCM) learning rule [5].

3.1. Anti-Oja rule

Oja's learning rule, proposed by Erkki Oja, is a model of how neurons in the brain or in artificial neural networks alter the strength of connections, or learn, over time. It is a modification of the standard Hebb's rule that, through multiplicative normalization, solves the stability problems and generates an algorithm for extracting principal components of an input vector [23]. This is a computational form of an effect that is believed to happen in biological neurons.

Hebb suggested his 'Hebbian' learning postulate in his book *The Organization of Behavior* [10] based on two principal rules. It states that if two neurons, located at the opposite sides of a synapse, are activated concurrently (synchronously), then the strength of the synaptic weight will increase. If two neurons, located at the opposite sides of a synapse, are activated at different times (asynchronously), then the strength of the synaptic weight will decrease.

Please cite this article as: M.-H. Yusoff et al., Modeling neural plasticity in echo state networks for classification and regression, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.11.017

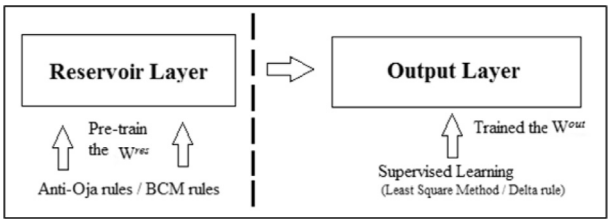

Fig. 2. An illustration of the diagram for pre-training of the reservoir weights and then update of readout weights.

The anti-Oja learning rule can be described as follows [3];

$$\Delta W_{\mathrm{kj}}(t) = \xi y_{\mathrm{k}}(t)[x_{\mathrm{j}}(t) - y_{\mathrm{k}}(t)W_{\mathrm{kj}}(t)] \tag{7}$$

where ΔW_{kj} is the change of the synaptic weight between the postsynaptic neuron, y_{k} , and the presynaptic neuron, x_{j}, at time, t. ξ is the learning rate. Note that Eq. (7) is a modified version of the anti-Hebbian rule by adding a forgetting factor to limit the growth of the synaptic weight to avoid the saturation of $\mathrm{W_{kj}}$.

3.2. BCM rule

The BCM rule [5] also follows the Hebbian learning principle, with a sliding threshold as a stabilizer function to control the synaptic alteration. The modification of this threshold controls the reduction and increase of the activity in the neurons. This leads to a self-regulation of the plasticity, which is believed to be able to improve the stability of learning.

This learning rule works on some temporal, moving average of pre- and post-synaptic activity. It also includes regulation that is based on the post-synaptic activity to reduce positive weight change when the level of change is a high. Using the BCM rule, the threshold and the speed of synaptic alteration become inversely proportional. When the speed of synaptic modification increases then the threshold is small, and decreases as the threshold increases.

The BCM rule has several variants and here we adopt the one suggested in [5] as follows:

$$\Delta W_{\mathrm{kj}}(t) = y_{\mathrm{k}}(y_{\mathrm{k}} - \theta_{\mathrm{M}})x_{\mathrm{j}}/\theta_{\mathrm{M}} \tag{8}$$

$$\theta_{\mathrm{M}} = E[y_k^2] = \sum p_k y_k^2 \tag{9}$$

where θ_M is the modification threshold of the postsynaptic neuron, $y_{k,}$ and p_k is the probability of choosing vector of y_k from the dataset, $E[.]$ is the temporal average, ΔW_{kj} is the adjustment of the synaptic weight between postsynaptic neuron, y_{k} and presynaptic neuron, $\mathrm{x_j}$ at time t.

4. Experimental setup

There are two possible options for performing plasticity on the weights within the reservoir and updating the readout weights. One is to carry out these two processes simultaneously, the other is to modify the reservoir weights first by applying plasticity rule and then update the readout weights by applying a supervised learning algorithm. We have tried implementing the first option however it did not work well due to the relatively higher computational cost as well as introducing a moving target for the supervised learning method. Consequently, we adopt the latter option, as illustrated in Fig. 2.

In the training phase, the ESN model was trained following three steps [12]. First, we randomly initialize W^{in}, W^{res} and W^{out} of the network. We use the standard normal distribution to initialize the value of the input weights, W^{in}. For predicting the Mackey-Glass time series, the values of W^{in} are scaled between −0.3 and 0.3, while for the Sunspots case study, the values of W^{in} are scaled to be between −0.05 and 0.05.

The weights inside the reservoir are randomly initialized within the range of −1.0 and 1.0. To make sure that the spectral radius (α) is less than 1, we divide the random value of W^{res} by the square root of the multiplication between the connection density and the number of reservoir neurons. Later, we normalize the value of spectral radius (α) [32].

In our experiments, the size of the input layer is variable depending on the number of history states (the input size in time series prediction) are used to predict the future. The size of the reservoir is set to 1000, that is, the reservoir contains 1000 neurons, unless otherwise specified. The reservoir topology is randomly initialized with a connectivity of 1% without self-feedback. The readout layer has only one neuron without output feedback. The readout weights are randomly initialized at the beginning and adapted when the learning process is implemented. The plasticity rule is applied for 100 iterations before the supervised learning is carried out.

The learning rate is set to 0.00001 for anti-Oja rule as recommended in [3]. When the BCM rule is applied, we use a moving average as the parameter to calculate the threshold.

According to [12], we need to have a washout time (T_0) so that the states of the network are independent of the initial states. That is to say, the states of the network will be updated for T_0 time steps before the plasticity rule is implemented.

Please cite this article as: M.-H. Yusoff et al., Modeling neural plasticity in echo state networks for classification and regression, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.11.017

In the online learning mode, 500 epochs of the supervised training is conducted to train the readout weights, W^{out}. Various learning rates have been tested to examine the influence of the learning rate on the performance.

In the experiments, we use the root mean square error (RMSE) as the error measurement for evaluating the offline and online learning performance.

After the training phase is complete, the network is tested on unseen data to validate its performance.

For the classification problems, we use the same setup for initializing all the weights of the ESN. Unlike the time series prediction tasks, we use the classification accuracy percentage for measuring the performance. We use the winner-takes-all principle to determine the predicted label.

4.1. Prediction of the Mackey–Glass time series

In this study, we use Mackey–Glass (MG) time series dataset for evaluating the performance of the ESNs for prediction tasks, as many studies in the literature showed that ESNs work is well suited for time series prediction compared to other methods [11,14]. We use delayed differential equations to generate MG time series dataset as follows:

$$\delta x/\delta t = [ax(t-\tau)/1 + x(t-\tau)10] - -bx(t) \tag{10}$$

where $x(t)$ is the value of the time series at time t, τ is set to 17, parameter $a = 0.2$, $b = 0.1$ and the step size is 0.1. We generate 12,000 data points using the MG system. The first 6000 points are used for the training and the remaining 6000 are used for the testing. Note that the first 1000 data points from the training set are used for washout only. ESNs with 1000 reservoir neurons are trained using two-step and five-step inputs, respectively, for prediction tasks. Input weights are initialized using the method explained above and the size of spectral radius is re-scaled and controlled to between 0.70–0.79. We use the root mean square error (RMSE) for evaluation of the results.

4.2. Prediction of the sunspot time series

In this study, we used the smoothed monthly sunspots dataset acquired from National Geophysical Data Center (NGDC) as the time-series data to predict, which contains 3132 samples of sunspots activity numbers from January 1750 until December 2010 [20,21,27]. The data consists of the year taken, months and sunspot group numbers.

For the input and output of the network, we follow the setup in [15] by using the series of sunspot numbers data ($d(t)$, $d(t-1)\ldots$, $d(t-K+1)$) as the input size of K to predict one step ahead of the input series data as the output of the model ($d(t+1)$) for time step t. We pre-process the data by normalizing each of the tuples according to the highest value of their elements. An input size of 15, 25 and 35 has been tested, respectively. We use 2100 time steps to train the network, and 1000 time steps for testing.

4.3. Additional time series datasets

We also used five additional datasets in our studies, including fluid flow in pipe, heat exchanger, magnetic levitation, pollution mortality, and S & P 500 U.S. financial datasets. For each dataset, 70% of the data are used for training and 30% for testing. The fluid flow in pipe dataset contains 1100 time steps and the ESN is trained to predict fluid flow from a pipe from the percentage of valve openings. For the heat exchanger dataset, we want to predict the outlet liquid temperature of a liquid-saturated steam heat exchanger, using past outlet liquid temperatures and/or the liquid flow rate. The total number of time steps in this dataset is 4000. The magnetic levitation dataset is used to predict the vertical position of a levitated magnet from the past values of its position and a control current through an electromagnet over which the levitated magnet is suspended. This dataset has 4001 time steps. The pollution mortality dataset is used to predict mortality due to pollution which has three kinds of mortality including total mortality, respiratory mortality or cardiovascular mortality. This dataset has 219 time steps, each containing eight measurement inputs, which are temperature, relative humidity, carbon monoxide, sulfur dioxide, nitrogen dioxide, hydrocarbons, and ozone particulates. Finally, the S & P 500 U.S. financial dataset has 930 time steps. This dataset is used to predict the future stock market price in U.S.

4.4. Classification of breast cancer and adult census income

In this study, we use Breast Cancer and Adult Census Income dataset as classification tasks for detailed comparisons, as these problems have widely been used as benchmark problems to evaluate classification performance of neural networks.

The Breast Cancer dataset has 699 samples, each consisting of nine features (inputs) and two classes, i.e., benign and malignant. In this task, we want to classify the type of the breast cancer either to be benign or malignant. Details of the dataset are listed in Table 1.

We normalize the dataset between 0 and 1 before training the ESN. We calculate the accuracy percentage to compare the ESN without applying plasticity to the one using plasticity to adapt the reservoir weights. In this experiment, we divide the dataset into two groups, one containing 519 samples for training and the other 180 samples for testing.

The Adult Census Income dataset contains 48,842 samples, each having 14 attributes. We eliminate a few samples with missing attributes and use only 32,728 samples, which are split into two groups, 70% for training and 30% for testing. For this

Please cite this article as: M.-H. Yusoff et al., Modeling neural plasticity in echo state networks for classification and regression, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.11.017

Table 1
Online learning (Input size =15).

η	$RMSE_{train}$			$RMSE_{test}$		
	No plasticity	anti-Oja	BCM	No plasticity	anti-Oja	BCM
0.01	0.0147	0.0191	0.0150	0.0184	0.0255	0.0367
0.07	0.0092	0.0093	0.0099	0.0112	0.0134	0.0843
0.10	0.0086	0.0086	0.0095	0.0103	0.0123	0.1057
0.50	0.0065	0.0065	∞	0.0081	0.0125	∞
0.60	0.0064	0.0063	∞	0.0081	0.0136	∞

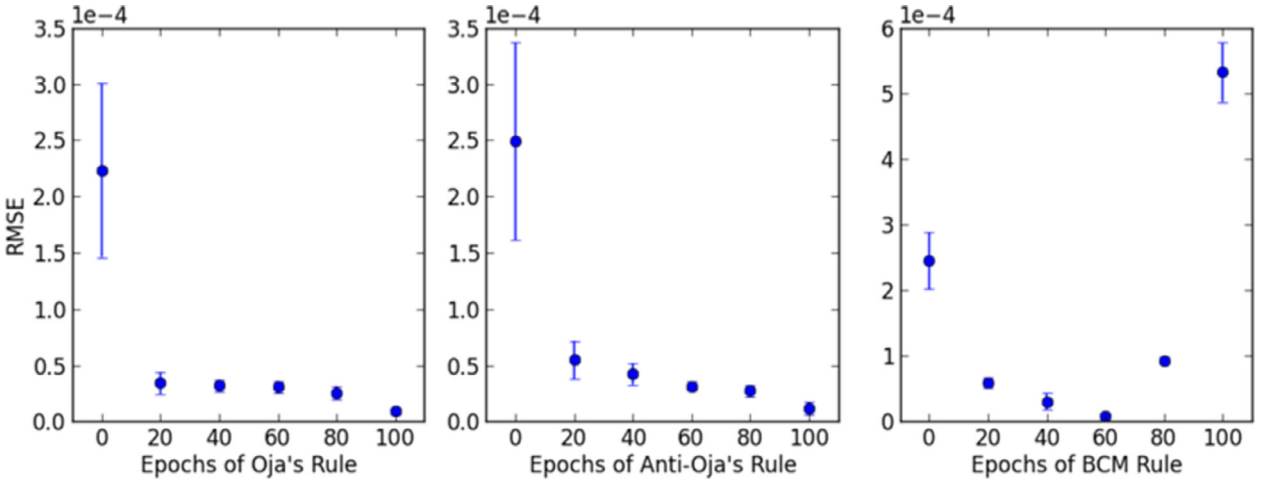

Fig. 3. Learning convergence of plasticity for the Mackey Glass prediction task.

problem, we want to classify whether the income of the person exceeds USD 50,000 per year or lower than USD 50,000. This is a binary classification problem.

4.5. Additional classification datasets

We also used four additional datasets, including Iris flowers, Glass (type of glass), Thyroid and Wine vintage dataset for comparing the classification performance. Similarly, 70% of the data are used for training and 30% for testing. For the Iris flowers dataset, we want to classify the species of Iris flower which contains four attribute inputs including Sepal length (cm), Sepal width (cm), Petal length (cm) and Petal width (cm). This dataset has 1000 samples and 3 target classes. For the Glass dataset, the classifier is required to classify the type as either window or non-window, depending on the glass chemistry. This dataset has two classes and nine attributes, which are Refractive index, Sodium (unit measurement: weight percent in corresponding oxide), Magnesium, Aluminum, Silicon, Potassium, Calcium, Barium and Iron. This dataset contains 214 glass samples. The Thyroid dataset has 7200 samples and each sample is referred to a clinic as normal, hyperfunction or subnormal functioning. It has 21 input variables and three target classes. For the Wine vintage dataset, we want to classify wines from three wineries in Italy based on constituents found through chemical analysis. This dataset has 178 samples, each of which containing 13 attributes including Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium, Total phenols, Flavanoids, Nonflavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines and Proline. This dataset has three target classes.

5. Experimental results

In this study, we will use the above-described two time series prediction problems and two classification tasks for evaluating the performance of the ESN models under comparison.

5.1. Learning convergence of plasticity

An initial set of experiments is undertaken to determine an appropriate number of iterations to apply plasticity for before it can be said to have converged in its learning. Fig. 3 shows the reduction in RMSE on the Mackey Glass data set for each of the plasticity rules: Anti-Oja, BCM, as well as the Hebbian version of Oja's rule.

It can be seen from Fig. 3, that plasticity mostly improves performance in the first 50 iterations. For the two variants of Oja's rule improvement slowly continues up to 100 iterations while the BCM rule dramatically degrades the performance after 60

Please cite this article as: M.-H. Yusoff et al., Modeling neural plasticity in echo state networks for classification and regression, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.11.017

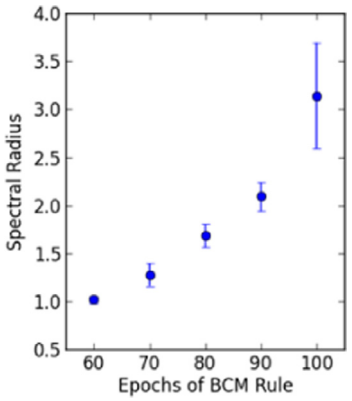

Fig. 4. Increase in spectral radius under BCM plasticity.

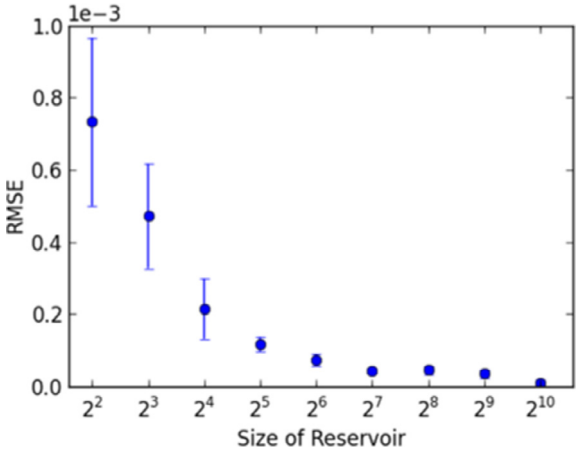

Fig. 5. Decrease in RMSE as reservoir size increases.

iterations. Therefore for the remainder of the results in this study, we select 100 iterations for Anti-Oja's rule and 60 for the BCM rule.

It is interesting to note that Hebbian and Anti-Hebbian variants of Oja's rule perform very similarly. This suggests that the improvement brought by the application of plasticity does not specifically depend on associative learning or decorrelation in the reservoir. Rather, it may be a more general principle of plasticity to allow the synaptic weights to adapt to the input structure.

To explain the increase in RMSE when the BCM rule is applied for more than 60 iterations, it is possible to analyze the change in spectral radius. Fig. 4 shows how the BCM rule dramatically drives the spectral radius of the weight matrix greater than 1.

By comparing Fig. 4 with the increase in RMSE in Fig. 3 for the BCM rule, we can see that the increase in error is due to the increase in spectral radius of the weight matrix. Later, in Section 5.6, a separate experiment is made to link these two quantities and determine an optimal setting.

5.2. Impact of reservoir size

Now, we investigate the influence of the reservoir size on the learning performance. Fig. 5 shows the results of testing an ESN with reservoir sizes between four and 1024 neurons. While low sizes of the reservoir give surprisingly good performance, testing error dramatically decreases up to about 64 neurons. Variability, due to the initial random weight values, also decreases as would be expected. As the size of 1000 neurons provides the best performance, that is the reservoir size we select for the remaining experiments.

5.3. Instability of online learning

The same experiments are now run using the online learning mode to determine the competitiveness of this configuration. In this mode, we use the delta rule to train the readout weights of the network after we implement the plasticity rule on the reservoir weights. The readout weights are then trained for 500 epochs using the delta rule. We used different learning rates (η)

Please cite this article as: M.-H. Yusoff et al., Modeling neural plasticity in echo state networks for classification and regression, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.11.017

Table 2
Online learning (Input size = 25).

η	$RMSE_{train}$			$RMSE_{test}$		
	No plasticity	anti-Oja	BCM	No plasticity	anti-Oja	BCM
0.01	0.0177	0.0193	∞	0.0238	0.0217	∞
0.07	0.0108	0.0110	∞	0.0175	0.0129	∞
0.10	0.0096	0.0097	∞	0.0168	0.0115	∞
0.50	0.0062	0.0060	∞	0.0144	0.0074	∞
0.60	0.0061	0.0058	∞	0.0142	0.0072	∞

Table 3
Online learning (Input size = 35).

η	$RMSE_{train}$			$RMSE_{test}$		
	No plasticity	anti-Oja	BCM	No plasticity	anti-Oja	BCM
0.01	0.0185	0.0191	0.0242	0.0213	0.0223	0.0488
0.07	0.0121	0.0114	0.0133	0.0142	0.0141	0.0371
0.10	0.0107	0.0100	0.0122	0.0128	0.0129	0.0377
0.50	0.0068	0.0122	∞	0.0102	0.0127	∞
0.60	0.0092	0.0520	∞	0.0104	0.0671	∞

Table 4
Spectral radius of the model using input size 15, 25 and 35.

Input size	Spectral radius		
	No plasticity	anti-Oja	BCM
15	0.7968	0.9222	1.0118
25	0.7960	0.9545	4.6825
35	0.7936	0.8798	1.0189

Table 5
The spectral radius for reservoir size = 1000.

Input size	Spectral radius		
	No plasticity	anti-Oja rule	BCM rule
2	0.795	0.814	0.861
5	0.800	0.826	0.851

to examine the influence of the learning rate on the learning performance. In these experiments, the reservoir size is fixed to 1000.

From Tables 1–3, we can see that the prediction error decreases when the learning rate increases. However, for an input size of 35, the errors become larger in case the anti-Oja rule is applied, and the output even diverges when the BCM rule is applied when the learning rate is 0.50 and 0.60, respectively.

For an input size of 25, we can see that the ESN fails to learn the target signal when the BCM rule is implemented in the network and the error is again divergent. As can be seen from Table 4, for a given size of 25, the spectral radius becomes 4.6825 when the BCM rule is applied. By contrast, the spectral radius is 0.9545 when the anti-Oja rule is applied. The previous observations suggest that the delta rule may not converge when the spectral radius is greater than 1, and could be a source of its instability.

Overall, it is clear that the performance results for the online learning model are significantly lower than the offline learning rule. Furthermore, online learning with the delta rule has proved unstable, with many experiments failing to converge. For these reasons, we select offline learning with LSE for the remainder of the experiments on other data sets in this study.

5.4. Time series prediction results

The results on the Mackey–Glass Time Series prediction using the offline learning are presented in Tables 5 and 6.

We used the BCM rule and anti-Oja rules, respectively, to update the reservoir weight of the ESN, in order to investigate the influence of the synaptic plasticity adaptation on the performance of ESNs. From Table 4, we can see that both the anti-Oja rule and the BCM rule are able to enhance the learning performance of ESNs. However, the test errors are smaller than the training errors. This is because the number of learning samples in the training set is slightly lower than in the test set as we take out 1000

Please cite this article as: M.-H. Yusoff et al., Modeling neural plasticity in echo state networks for classification and regression, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.11.017

Table 6
RMSE on MG time series for tau = 17 & reservoir size = 1000.

Method	Input size			
	2		5	
	$RMSE_{train}$	$RMSE_{test}$	$RMSE_{train}$	$RMSE_{test}$
No plasticity	1.230 E−6	1.695 E−6	9.819 E−7	9.811 E−7
anti-Oja	4.941 E−7	4.941 E−7	2.305 E−7	2.244 E−7
BCM	7.107 E−7	6.987 E−7	1.871 E−7	1.848 E−7

Table 7
The RMSE on sun spot data using input size 15, 25 and 35.

Input size	$RMSE_{train}$			$RMSE_{test}$		
	No plasticity	Anti-Oja	BCM	No plasticity	Anti-Oja	BCM
15	0.0097	0.0022	0.0052	0.0627	0.0161	0.0441
25	0.0058	0.0020	0.0059	0.0826	0.0217	0.0106
35	0.0139	0.0114	0.0062	0.0507	0.0466	0.0269

Table 8
Performance results (RMSE) for other prediction tasks.

Dataset	Offline learning algorithm for regression					
	Fixed network		anti-Oja		BCM	
	$RMSE_{train}$	$RMSE_{test}$	$RMSE_{train}$	$RMSE_{test}$	$RMSE_{train}$	$RMSE_{test}$
1. Fluid flow in pipe	0.0123	0.0172	0.0020	0.0152	0.0072	0.0123
2. Heat exchanger	5.21 E-2	6.33 E-2	4.57 E-2	4.89 E-2	3.99 E-2	4.15 E-2
3. Magnetic levitation	9.11 E-2	1.25 E-1	6.48 E-2	7.55 E-2	5.94 E-2	6.29 E-2
4. Pollution mortality	0.0796	0.0840	0.0571	0.0633	0.0529	0.0599
5. S&P500 (US financial)	0.0565	0.0666	0.0412	0.0468	0.0332	0.0434

data points from the training set for washout. We also note that the best performance is achieved when the input size is set to five with the BCM rule model is applied.

The results on the sunspot prediction data using offline learning are presented in Table 8. To apply the plasticity rules, the initial spectral radius is set to be within 0.79∼0.83 to make sure that the network has the echo state property (spectral radius <1) [11].

From Table 7, we can see that by applying the anti-Oja learning rule and the BCM rule in the reservoir, the errors have been reduced compared to the network model without plasticity being applied. In addition, one interesting observation we can make is that the spectral radius value is increased when the weights inside the reservoir are updated using plasticity. We can see that the size of inputs has considerable influence on the learning performance and an increase in input size degrades the learning performance.

Fig. 6(a) and (b) shows the prediction results on the training and test data, respectively, for an input size of 25 and when the BCM rule is applied in offline learning mode.

Summarizing the above results on the sunspot dataset, we can conclude that applying plasticity to the reservoir is able to enhance the learning performance of the ESN in the offline learning mode, while the learning performance becomes unstable in case the online learning algorithm mode is adopted. The state of the ESN is very likely to become divergent due to the fact that the spectral radius is driven larger than 1 by plasticity. For this reason, the ESN model without plasticity produces better results in the online learning mode than in the offline learning mode. Even though the anti-Oja learning rule obtains the smallest test error in the online learning mode, it does not improve the result in the fixed network.

We also implement the model using other datasets such as Fluid flow in pipe, Heat exchanger, Magnetic levitation, Pollution mortality and S&P 500 financial dataset. The results are shown below. In this experiment, we use offline learning algorithm.

Based on the results in Table 8, we can see that the performance of prediction is improved when we implement the plasticity rules in the network. In the offline learning mode, applying the BCM rule has resulted in better results compared to the anti-Oja rule. All the values above are the average of 10 runs, which standard deviation did not exceed 0.009066.

5.5. Classification results

Table 9 lists the performance values of all six classification datasets tested with each plasticity rule. We can see that applying the BCM model produces better performance for classification than the anti-Oja rule which in turn is an improvement over a static ESN.

Please cite this article as: M.-H. Yusoff et al., Modeling neural plasticity in echo state networks for classification and regression, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.11.017

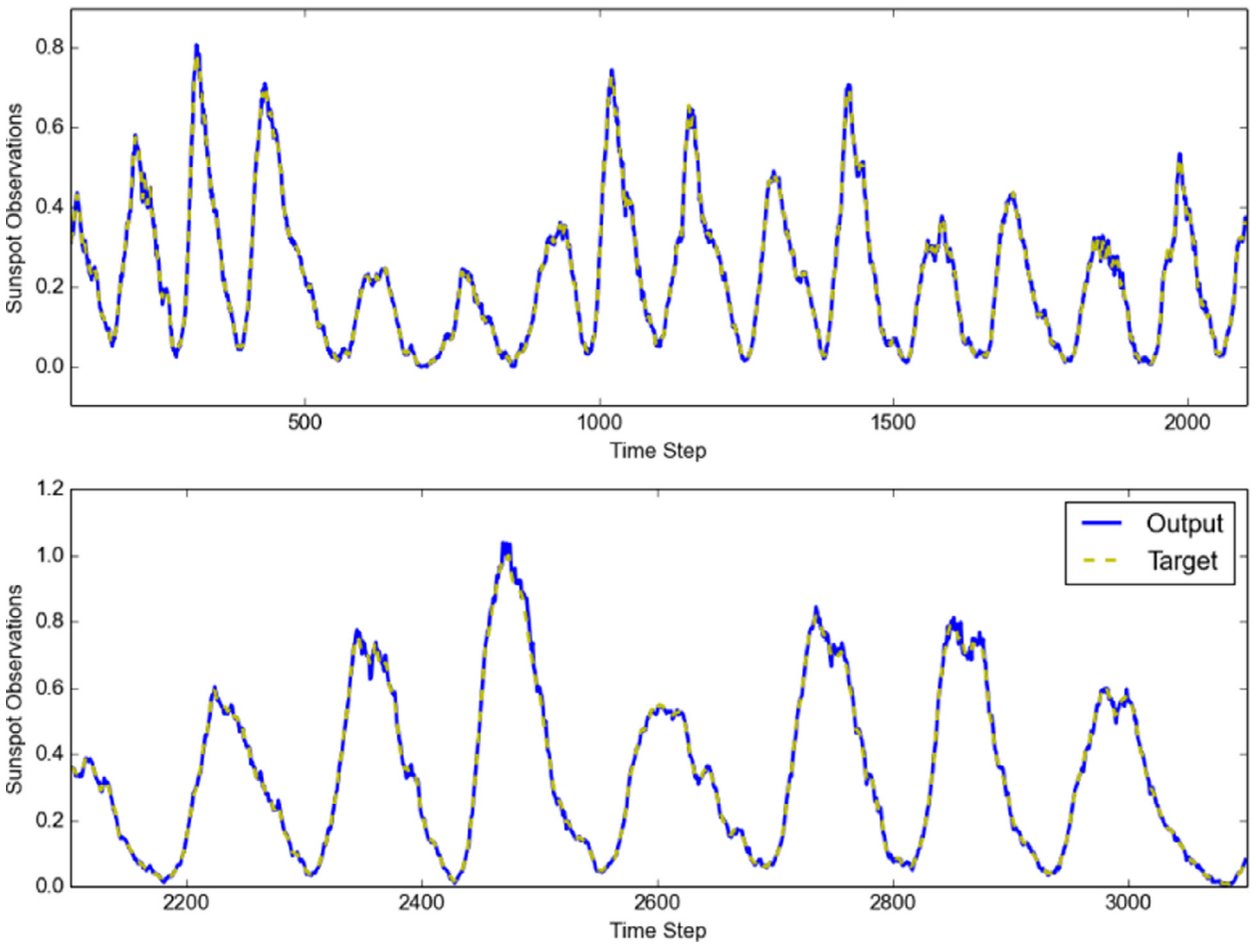

Fig. 6. Offline learning results. Top: Training phase, Bottom: Testing phase.

Table 9
Performance results (accuracy) for classification tasks.

Dataset	Offline learning algorithm for classification					
	Fixed network		Anti-Oja		BCM	
	% training	% testing	% training	% testing	% training	% testing
1. Breast cancer	0.9637	0.9512	0.9742	0.9639	0.9818	0.9783
2. Adult census income	0.8645	0.8549	0.8963	0.8831	0.9128	0.8977
3. Iris flower	0.9731	0.9568	0.9912	0.9742	0.9946	0.9854
4. Glass (type of glass)	0.9685	0.9572	0.9774	0.9624	0.9815	0.9783
5. Thyroid	0.9274	0.9111	0.9412	0.9286	0.9478	0.9310
6. Wine vintage	0.9572	0.9487	0.9593	0.9514	0.9633	0.9583

From Table 9, we can see that the ESN model with the BCM rule outperforms the model with the anti-Oja rule in the classification tasks. All the values above are also the average of 10 runs, which standard deviation did not exceed 0.005772.

5.6. Effect of spectral radius on performance

In this set of experiments, we investigate the influence of different values of spectral radius in the reservoir weights on the learning performance. More specifically, we manually set the spectral radius by increasing increments and measure the testing performance of the model. We use Mackey–Glass time series data in this study. Here, we use the ESN model without plasticity applied to the reservoir weights as doing so could change the values of spectral radius from our manually selected increments. The simulation is performed ten times for each reading to get the average of the test error. We randomly initialize the input

Please cite this article as: M.-H. Yusoff et al., Modeling neural plasticity in echo state networks for classification and regression, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.11.017

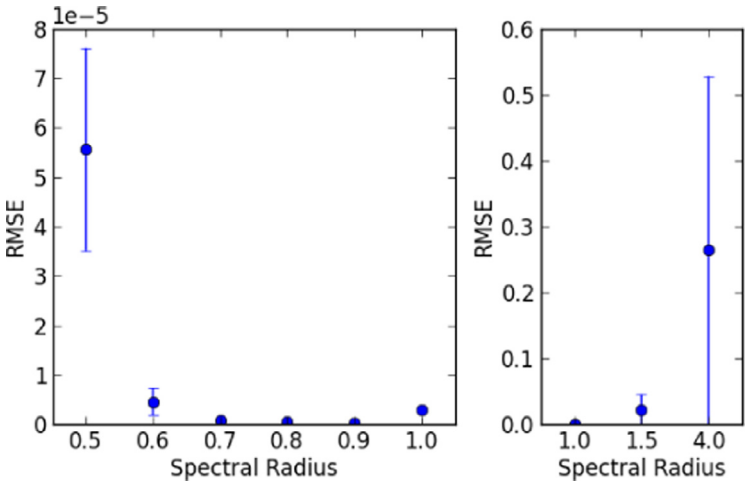

Fig. 7. The Mean Testing error versus Spectral Radius.

weights, reservoir weights and the readout weights at the beginning of the simulation. The reservoir weights are scaled to control the spectral radius. The results are presented in Fig. 7.

In Fig. 7, we can see that the increment of the spectral radius first decreases the testing error while it still less than one. However, when the spectral radius is larger than unity (spectral radius, $\alpha > 1$), the testing error is significantly increased thus the performance become worst. This result shows that the size of spectral radius is more sensitive to the performance of the task when the value is shifting and approaching the unity value. For the ESN model, values between 0.7 and 0.9 seem to be optimal.

6. Conclusions

In this work, two neural plasticity rules have been employed for updating the weights inside the reservoir of the ESN model for time series prediction and classification. Our results indicate that the BCM rule is more effective than the anti-Oja rule in enhancing the learning performance of the ENS, especially in the offline learning mode. In case the online learning mode is used, the anti-Oja rule is more stable than the BCM rule, although neither of them is able to considerably improve the learning performance.

The instability of the online learning mode can partly be attributed to the increased spectral radius of the reservoir. This happens in our experiments mainly in the online learning mode. Understanding this learning behavior will be our future work.

In addition, new effective plasticity rules will be studied and applied in semi-supervised learning environments, which hopefully will be more effective and more reliable for enhancing the learning performance of ESNs. We are also interested in investigating multi-layer ESNs, as these networks can produce richer dynamics and are expected to significantly improve the performance of ESNs for regression and classification tasks [34].

Acknowledgments

This research was supported in part by the University Malaysia Kelantan (UMK) and the Engineering and Physical Sciences Research Council (EPSRC), UK, Project No. EP/M017869/1.

References

[1] M. Alhamdoosh, D. Wang, Fast decorrelated neural network ensembles with random weights, Inf. Sci. (Ny) 264 (2014) 104–117.
[2] J. Š. Babinec, Pospíchal, optimization in Echo state neural networks by Metropolis algorithm, in: Proceedings of the 10th International Conference Soft Computing, Mendel, 2004, pp. 155–160.
[3] J. Š. Babinec, Pospíchal, Improving the prediction accuracy of echo state neural networks by anti-Oja's learning, Artif. Neural Networks–ICANN 2007, Springer, 2007, pp. 19–28.
[4] F. Cao, D. Wang, H. Zhu, Y. Wang, An iterative learning algorithm for feedforward neural networks with random weights, Inf. Sci. (Ny) 328 (2016) 546–557.
[5] G.C. Castellani, N. Intrator, H. Shouval, L.N. Cooper, Solutions of the BCM learning rule in a network of lateral interacting nonlinear neurons, Netw. Comput. Neural Syst. 10 (1999) 111–121.
[6] J. Chrol-Cannon, Y. Jin, Computational modeling of neural plasticity for self-organization of neural networks, BioSystems 125 (2014) 43–54.
[7] J. Chrol-Cannon, Y. Jin, Learning structure of sensory inputs with synaptic plasticity leads to interference, Front. Comput. Neurosci. 9 (2015) 1–13.
[8] S.A. Van De Geer, Least-Squares Estimation, John Wiley & Sons, Ltd, 2008.
[9] M.D. Haene, Toward unified hybrid simulation techniques for spiking neural networks, Neural Comput. 26 (1079) (2014) 1055–1079.
[10] D.O. Hebb, The organization of behavior: A neuropsychological theory, Psychology Press, 2005.
[11] H. Jaeger, The "echo state" approach to analysing and training recurrent neural networks-with an erratum note, Bonn, Ger. Ger. Natl. Res. Cent. Inf. Technol. GMD Tech. Rep. 148 (2001) 34.
[12] H. Jaeger, Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach, GMD-Forschungszentrum Informationstechnik (2002).

[13] H. Jaeger, Adaptive nonlinear system identification with echo state networks, in: AdvacesNeural Information Processing Systems, 2002, pp. 593–600.
[14] H. Jaeger, H. Haas, Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication, Science 304 (2004) 78–80.
[15] H. Jaeger, M. Lukosevicius, D. Popovici, U. Siewert, Optimization and applications of echo state networks with leaky-integrator neurons, Neural Netw 20 (2007) 335–352.
[16] N. Kasabov, E. Capecci, Spiking neural network methodology for modelling, classification and understanding of EEG spatio-temporal data measuring cognitive processes, Inf. Sci. (Ny) 294 (2015) 565–575.
[17] M. Lukoševičius, A practical guide to applying echo state networks, Neural Networks: Tricks of the Trade, Springer, 2012, pp. 659–686.
[18] M. Lukoševičius, H. Jaeger, Reservoir computing approaches to recurrent neural network training, Comput. Sci. Rev 3 (2009) 127–149.
[19] M. Lukoševičius, H. Jaeger, B. Schrauwen, Reservoir computing trends, KI-K{ü}nstliche Intelligenz 26 (2012) 365–371.
[20] NASA, {Solar Cycle} Prediction, 2013. http://solarscience.msfc.nasa.gov/predict.shtml (accessed September 15, 2013).
[21] National Center for Environmental Information NOAA, {Sunspot} Number, (2013).
[22] K. Neumann, C. Emmerich, J.J. Steil, Regularization by intrinsic plasticity and its synergies with recurrence for random projection methods, (2012).
[23] E. Oja, Simplified neuron model as a principal component analyzer, J. Math. Biol 15 (1982) 267–273.
[24] J.C. Pemberton, J.J. Vidal, When is the generalized delta rule a learning rule? a physical analogy, in: neural networks, in: IEEE International Conference, 1988, 1988, pp. 309–315.
[25] F. Schwenker, A. Labib, Echo state networks and neural network ensembles to predict sunspots activity, Network 3 (2009) 2.
[26] M.D. Skowronski, J.G. Harris, Minimum mean squared error time series classification using an echo state network prediction model, in: circuits Syst. 2006, in: ISCAS 2006. Proceedings 2006 IEEE International Symposium, 2006, p. 4.
[27] Solar Influences Data Analysis Center (SIDC), {SilSo} Data Files, (2013).
[28] Q. Song, Z. Feng, Effects of connectivity structure of complex echo state network on its prediction performance for nonlinear time series, Neurocomputing 73 (2010) 2177–2185.
[29] J.J. Steil, Backpropagation-decorrelation: online recurrent learning with O (N) complexity, in: neural networks, 2004, in: Proceedings of the 2004 IEEE International Jt. Conference, 2004, pp. 843–848.
[30] J.J. Steil, Online reservoir adaptation by intrinsic plasticity for backpropagation-decorrelation and echo state learning, Neural Netw 20 (2007) 353–364.
[31] G.K. Venayagamoorthy, B. Shishir, Effects of spectral radius and settling time in the performance of echo state networks, Neural Netw 22 (2009) 861–863.
[32] W. Yuanbiao, J. Ni, X. Zhiping, Effects of Spectral Radius on Echo-State-Network's Training, in: internet comput. sci. eng. (ICICSE), in: 2009 Fourth International Conference, 2009, pp. 102–108.
[33] F. Zhai, X. Lin, Z. Yang, Y. Song, Financial time series prediction based on echo state network, in: 2010 Sixth International Conference Natural Computing, 2010, pp. 3983–3987.
[34] X. Zhu, A.B. Goldberg, Introduction to semi-supervised learning, Synth. Lect. Artif. Intell. Mach. Learn 3 (2009) 1–130.

Please cite this article as: M.-H. Yusoff et al., Modeling neural plasticity in echo state networks for classification and regression, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.11.017