Modeling Neural Plasticity in Echo State Networks for Time Series Prediction

Mohd-Hanif Yusoff  
Department of Computing  
Faculty of Engineering and Physical Sciences  
University of Surrey, Guildford  
GU2 7XH, Surrey, UK  
m.yusoff@surrey.ac.uk

Yaochu Jin  
Department of Computing  
Faculty of Engineering and Physical Sciences  
University of Surrey, Guildford  
GU2 7XH, Surrey, UK  
yaochu.jin@surrey.ac.uk

Abstract—In this paper, we investigate the influence of neural plasticity on the learning performance of echo state networks (ESNs) and supervised learning algorithms in training readout connections for two time series prediction problems including the sunspot time series and the Mackey Glass chaotic system. We implement two different plasticity rules that are expected to improve the prediction performance, namely, anti-Oja learning rule and the Bienenstock-Cooper-Munro (BCM) learning rule combined with both offline and online learning of the read-out connections. Our experimental results have demonstrated that the neural plasticity can more significantly enhance the learning in offline learning than in online learning.

Index Terms—Echo State Networks; Synaptic Plasticity; Learning algorithms

I. INTRODUCTION
Echo State Networks (ESNs) introduced a new paradigm in training recurrent neural networks (RNNs) for solving supervised problems. In comparison to the feed forward networks, RNNs have richer dynamics [1]. When given an input signal, via the recurrent connection pathways, an RNN preserves a nonlinear transformation of the input history in its internal state. With such dynamic memory, it offers some advantages to RNNs for temporal information processing. However, despite the rich dynamics of RNNs, effective training of RNNs becomes very challenging [2].

The ESN provides a new and promising framework for training RNNs. Learning in ESNs can be implemented by using a linear regression algorithm [3]. 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. One unique property of ESN lies in the way in which the network updates its weights. The modification of weights only involves the readout connections, whilst other weights are not plastic.

The advantage of efficient learning has made ESN to become popular and widely used for solving different learning problems such as pattern classification, e.g. [4], [5] and time-series prediction, e.g. [6],[7]. One weakness of the ESN is that it is not straightforward to improve the learning performance [2]. One potential solution is to modulate the dynamics of its reservoir activity.

Computational modeling of neural plasticity and its role in self-organization of artificial neural network models have been widely investigated [30]. However, not much research work has been reported on the influence of neural plasticity on learning performance of reservoir based neural network models, with few exceptions [2]. In this paper, we investigate the interaction between the plasticity in an ESN reservoir and the plasticity in its readout connections. Two plasticity rules, the anti-Oja rule [2] and BCM rule [8] have been implemented in the ESN for unsupervised learning in the reservoir. Once the unsupervised learning of the weights in the reservoir is complete, a supervised learning algorithm is applied to the read-out weights. Here, two supervised learning approaches have been investigated, one on-line approach and one off-line approach. By combining two plasticity rules with two supervised learning approaches, four different learning scenarios have been studied on two time series prediction problem, the sunspot time series and the Mackey-Glass time series. Our results indicate that the plasticity rules are efficient in enhancing the subsequent supervised learning of the read-out connections if the supervised learning is conducted off-line.
II. ECHO STATE NETWORK

A. The ESN Architecture

The network consists of K input neurons, N inner (i.e. reservoir) neurons and L readout neurons. The synaptic connections from the input neurons to the neurons in the reservoir, within the reservoir, neurons in the reservoir to the readout neuron are stored in the weights matrix \( W^i \), \( W^{res} \) and \( W^{out} \), respectively, as illustrated in Fig. 1.

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

\[
x(t+1) = f(W^i u(t+1) + W^{res} x(t))
\]

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

The readout, \( y \) is computed using (2):

\[
y(t+1) = W^{out} (u(t+1), x(t+1), y(t))
\]

where \( u(t+1), x(t+1), y(t) \) is the concatenation of the input, internal (reservoir), and previous output activation vectors [3].

![Fig. 1: The basic architecture of Echo State Network model adapted from [3].](image)

B. Learning of readout neuron

In this study, we consider both offline [3] and online learning [29] for training the readout weights. The plasticity rules are only used to adapt the weight connections in the reservoir. The readout weights are trained separately using a supervised learning algorithm once the unsupervised learning is complete.

In the offline learning mode, the readout weights are updated using all training data. In this study, we used the least square estimation (LSE) method [27] to calculate the best weight values for the readout connection that minimize the difference between the desired (target) output signal and the real output, in a single step computation as follows:

\[
E(W^{out}) = \frac{1}{T} \sum_{t=1}^{T} \| y^{\text{desired}}(t) - y(t) \|^2
\]

where \( E \) is the error on the training data for the given weights \( W^{out} \), \( y^{\text{desired}} \) is the desired output, and \( y \) is the actual output of the network. The LSE can be described by:

\[
W^{out} = (R R^T)^{-1} R y^{\text{desired}}(t)
\]

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 computationally a very simple method for training the read-out 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 also known as batch learning method [27].

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 one by one in a random order. We used the delta rule for online learning, which is a gradient descent method originally used for updating the weights of a single-layer perceptron model [63]. The Delta rule can be described as follows:

\[
\Delta W = \eta(y^{\text{desired}}(t) - y(t))(x(t))
\]

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

III. PLASTICITY RULES

The main goal of this study is to examine the effectiveness of implementing synaptic plasticity in the internal (reservoir) layer of an ESN. More specifically, we investigate the influence of synaptic plasticity on the learning performance of ESNs on time-series prediction.

In the default ESN model defined by [3], the supervised learning process on the read-out weights is performed without applying plasticity to the reservoir. The connectivity of the reservoir is randomly generated, as well as the connection strength of the connections within the reservoir. However, we hypothesize that the optimal reservoir connectivity depends on the problem to be learned. 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 structure 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 implement two different synaptic plasticity rules that are expected to improve the prediction performance. The two rules are: Anti-Oja learning rule [2] and the Bienenstock- Cooper-Munro (BCM) learning rule [8].

A. 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 connection, 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 principal components analysis [26]. This is a computational form of an effect which is believed to happen in biological neurons.

Hebb suggested the Hebbian learning rule in his book *The Organization of Behavior* [13] based on two principle rules. It says that if two neurons, located at the opposite sides of a synapse, are activated concurrently (synchronously), then the value 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 value of the synaptic weight will decrease.

For Anti-Oja Learning, the equation can be described as follows [2]:

\[
\Delta W_{kj}(t) = \xi y_k(t)x_j(t) - y_k(t)W_{kj}(t)
\]  

(7)

where \(\Delta W_{kj}\) is the adjustment of synaptic weight between postsynaptic neuron, \(y_k\), and presynaptic neuron, \(x_j\), at time, \(t\). \(\xi\) is the learning rate. Note that Equation (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 \(W_{kj}\).

B. BCM rule

The BCM rule [8] also follows the Hebbian learning principle, with a sliding threshold as the stabilizer function to control the synaptic alteration. The modification of threshold controls the reducing and increasing activity of the neurons. This leads to a self-regulation of the plasticity and hopefully better stability.

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 there is a high level. Using the BCM rules, the threshold and 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.

For BCM rules, we used the one suggested in [8] as follows:

\[
\Delta W_{kj}(t) = y_k(y_k - \theta_M)x_j / \theta_M
\]

(8)

\[
\theta_M = E[y_k^2] = \sum p_k y_k^2
\]

(9)

where \(\theta_M\) is the modification threshold of postsynaptic neuron, \(y_k\) and \(p_k\) is the probability of choosing vector of \(y_k\) from the dataset, \(E[.]\) is the temporal average, \(\Delta W_{kj}\) is the adjustment of synaptic weight between postsynaptic neuron, \(y_k\), and presynaptic neuron, \(x_j\), at time, \(t\).

IV. EXPERIMENTAL SETUP

There are two possible options for performing plasticity on the weights in the reservoir and updating the read-out weights. One is to carry out these two processes simultaneously, the other is to modify the reservoir weights first and then update the read-out weights. In this work, we take the latter option, as illustrated in Fig. 2.

In the training phase, the ESN model was trained following three steps [9]. First, we initialized randomly the \(W^{in}\), \(W^{res}\) and \(W^{out}\) of the network. For input weights, \(W^{in}\), we used standard normal distribution to initialize the value. In Mackey-Glass time series experiment, the values of \(W^{in}\) are between -0.3 to 0.3 while in Sunspots experiment, the values of \(W^{in}\) are between -0.05 to 0.05.

In the reservoir, the weights are generated randomly with the range of -1 and 1. To make sure that the spectral radius (\(\alpha\)) is less than 1, we divide the random value of \(W^{res}\) with the square root of multiplication between connection density and the number of reservoir neuron.

In our experiments, the size of the input layer is variable depending on how many history states (input size) 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 random initialized with a connectivity of 1% without self-feedback. Meanwhile, the readout layer only has 1 neuron with no output feedback. The readout weight is initialized randomly 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.

We used the learning rate of 0.00001 for Anti-Oja rule as recommended in [2]. When the BCM rule is applied as the plasticity, we use a moving average as the parameter to calculate the threshold.

According to [9], we need to have a washout time (\(T_0\)) so that the states of the network are independent of the early states. That is to say, the states of the network will be updated for \(T_0\) time steps before the plasticity rule is implemented.

For the on-line learning, 500 epochs of supervised training is conducted to train the output weights, \(W^{out}\). Various learning rates have been tested to examine influence of the learning rate on the learning performance.

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

After the training phase is finished, the network is tested. We used the trained \(W^{in}\), \(W^{res}\) and \(W^{out}\) of the network for the testing phase based on the minimum value of RMSE produced by the trained network. We use a linear regression function to compute the readout.
V. EXPERIMENTAL RESULTS

In this study, we will use two time series prediction datasets, i.e., the Mackey Glass time series (MG) and Sunspot time series to evaluate the performance of ESN model with neural plasticity adaptation in the reservoir. These two test problems have widely been used as a benchmark problem for time series prediction [6].

A. Prediction of the Mackey-Glass Time Series

In this study, we use Mackey-Glass (MG) time series dataset for prediction task, as many studies in the literature showed that ESNs work well in performing prediction abilities compared to others approach [3, 11]. We use delayed differential equations to generate MG time series dataset following the equation:

\[
\frac{dx}{dt} = \left[ a x(t - \tau) / 1 + x(t - \tau) \right] - bx(t)
\]  

(10)

where \(x(t)\) is the value of time series at time \(t\), \(\tau\), is 17, parameter \(a = 0.2\), \(b = 0.1\) and the step size is 0.1. We have generated 12000 values using MG system. The first 6000 values were used for the training set and remaining 6000 values were used for testing. Initial 1000 from the training set were used only for washout. Models with 1000 hidden units were trained for the two-step and five-step input value prediction task. Input weights were initialized using the method explain in Section IV and the size of spectral radius is re-scaled and controlled to 0.70 ~ 0.79. We use the root mean square error (RMSE) to compare the models and results are shown in the Table I and Table II.

**TABLE I. THE SPECTRAL RADIUS FOR RESERVOIR SIZE = 1000**

<table>
<thead>
<tr>
<th>Input Size</th>
<th>Spectral radius</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No Plasticity</td>
</tr>
<tr>
<td>2</td>
<td>0.795</td>
</tr>
<tr>
<td>5</td>
<td>0.800</td>
</tr>
</tbody>
</table>

**TABLE II. RESULTS ON MG TIME SERIES FOR TAU = 17 & RESERVOIR SIZE=1000**

<table>
<thead>
<tr>
<th>Method</th>
<th>RMSE(_{\text{train}})</th>
<th>RMSE(_{\text{test}})</th>
<th>RMSE(_{\text{train}})</th>
<th>RMSE(_{\text{test}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>No plasticity</td>
<td>1.230 E-6</td>
<td>1.695 E-6</td>
<td>9.819 E-7</td>
<td>9.811 E-7</td>
</tr>
<tr>
<td>Anti-Oja</td>
<td>4.941 E-7</td>
<td>4.941 E-7</td>
<td>2.305 E-7</td>
<td>2.244 E-7</td>
</tr>
<tr>
<td>BCM</td>
<td>7.107 E-7</td>
<td>6.978 E-7</td>
<td>1.871 E-7</td>
<td>1.848 E-7</td>
</tr>
</tbody>
</table>

We used BCM rules and Anti-Oja rules, respectively to update the reservoir weight of ESN, in order to see the influence of the synaptic plasticity adaptation on the performance of ESN. From Table II, we can see that both the anti-Oja rule and the BCM rule are able to enhance the learning performance of the ESN. However, the test errors are smaller than the training errors. This happen because the number of learning samples in training set is slightly lower than in testing set as we take out 1000 values of training set for washout. We also note that the best performance was achieved when the input size is set to five with the BCM rules model in the reservoir network.

In the next, we investigate the influence of the reservoir size on the learning performance. To this end, the reservoir size was changed to 500. The results are presented in Tables III and IV, which are very similar to those when the reservoir size is 1000.

**TABLE III. THE SPECTRAL RADIUS FOR RESERVOIR SIZE = 500**

<table>
<thead>
<tr>
<th>Input Size</th>
<th>Spectral radius</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No Plasticity</td>
</tr>
<tr>
<td>2</td>
<td>0.785</td>
</tr>
<tr>
<td>5</td>
<td>0.772</td>
</tr>
</tbody>
</table>

**TABLE IV. RESULTS ON MG TIME SERIES FOR TAU = 17 & RESERVOIR SIZE=500**

<table>
<thead>
<tr>
<th>Method</th>
<th>RMSE(_{\text{train}})</th>
<th>RMSE(_{\text{test}})</th>
<th>RMSE(_{\text{train}})</th>
<th>RMSE(_{\text{test}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>No plasticity</td>
<td>1.531 E-6</td>
<td>1.536 E-6</td>
<td>9.611 E-7</td>
<td>9.534 E-7</td>
</tr>
<tr>
<td>Anti-Oja</td>
<td>4.480 E-7</td>
<td>4.458 E-7</td>
<td>7.308 E-7</td>
<td>7.394 E-7</td>
</tr>
<tr>
<td>BCM</td>
<td>8.449 E-7</td>
<td>8.373 E-7</td>
<td>1.589 E-7</td>
<td>1.564 E-7</td>
</tr>
</tbody>
</table>

B. Prediction of the Sunspot Time Series

In this study, we used the smoothed monthly sunspots dataset acquired from National Geophysical Data Center (NGDC) as a time-series data, which contain 3132 samples of sunspots activity numbers from January 1750 until December 2010[10]. The data consists of year taken, months and sunspot group numbers.

For the input and output of the network, we follow the setup in [6] by using the series of sunspot numbers data (d(t), d(t-1)….d(t+K)) 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)) in time step, t. We pre-processed the data by normalizing each of tuple with the highest value number of elements. An input size of 15, 25 and 35 has been tested, respectively. We use 2100 time steps to train the network while 1000 time steps for test.

The results on the sunspot prediction data using the offline learning without applying plasticity on the reservoir weights are presented in Table V.

**TABLE V. RESULTS FROM OUR EXPERIMENT FOR THE MODEL USING INPUT SIZE 15, 25 AND 35 WITHOUT PLASTICITY FOR OFFLINE LEARNING**

<table>
<thead>
<tr>
<th>Input Size</th>
<th>Offline Learning algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSE(_{\text{train}})</td>
</tr>
<tr>
<td>15</td>
<td>0.0097</td>
</tr>
<tr>
<td>25</td>
<td>0.0058</td>
</tr>
<tr>
<td>35</td>
<td>0.0139</td>
</tr>
</tbody>
</table>

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) [3]. The results using offline learning with plasticity are shown in Table VI.
The influence of the learning rate on the ESN's value can be seen clearly that the ESNs value is elastic to the BCM rule. When the Oja rule is applied, and the output even diverges when the input size 35, for cases where learning rate is 0.50 and 0.01. Based on the Tables VIII, IX and X, we can see that the error size of inputs has influence to the error produced by the network and an increase in input size degrades the learning performance.

From Table VI, we can see that by applying Anti-Oja learning rule and BCM rule in the reservoir, the errors have been reduced compared to the network with no plasticity. In addition, an interesting observation we can make is that the spectral radius value is increased when reservoir weights are updated using plasticity. We can see that the size of inputs has influence to the error produced by the network and an increase in input size degrades the learning performance.

<table>
<thead>
<tr>
<th>IS</th>
<th>RMSE\textsubscript{train}</th>
<th>RMSE\textsubscript{test}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No plasticity</td>
<td>Anti-Oja</td>
</tr>
<tr>
<td>15</td>
<td>0.0097</td>
<td>0.0022</td>
</tr>
<tr>
<td>25</td>
<td>0.0058</td>
<td>0.0020</td>
</tr>
<tr>
<td>35</td>
<td>0.0139</td>
<td>0.0114</td>
</tr>
</tbody>
</table>

Next, we run the experiment using online learning. For this approach, we use Delta Rule to train the readout weights of the network after we implement plasticity rule on the reservoir weights. The readout weights are then trained for 500 epochs using the Delta rule. We used different learning rates ($\eta$) to examine the influence of the learning rate on the learning performance. The reservoir size is fixed to 1000.

<table>
<thead>
<tr>
<th>Input Size</th>
<th>Spectral Radius</th>
</tr>
</thead>
<tbody>
<tr>
<td>No plasticity</td>
<td>Anti-Oja</td>
</tr>
<tr>
<td>15</td>
<td>0.7968</td>
</tr>
<tr>
<td>25</td>
<td>0.7960</td>
</tr>
<tr>
<td>35</td>
<td>0.7936</td>
</tr>
</tbody>
</table>

For an input size 25, we can see clearly that the ESN fails to learn the target signal when BCM rule is implemented in the network and the error is divergent. Actually, 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.

Fig. 3 and Fig. 4 show 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.

<table>
<thead>
<tr>
<th>$\eta$</th>
<th>RMSE\textsubscript{train}</th>
<th>RMSE\textsubscript{test}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No plasticity</td>
<td>Anti-Oja</td>
</tr>
<tr>
<td>0.01</td>
<td>0.0185</td>
<td>0.0191</td>
</tr>
<tr>
<td>0.07</td>
<td>0.0121</td>
<td>0.0114</td>
</tr>
<tr>
<td>0.10</td>
<td>0.007</td>
<td>0.0100</td>
</tr>
<tr>
<td>0.50</td>
<td>0.0068</td>
<td>0.0122</td>
</tr>
<tr>
<td>0.60</td>
<td>0.0092</td>
<td>0.0520</td>
</tr>
</tbody>
</table>

Based on the Tables VIII, IX and X, we can see that the error is decreased when we increase the learning rate. However, for the input size 35, for cases where learning rate is 0.50 and 0.60, respectively, the errors become larger in case the anti-Oja rule is applied, and the output even diverges when the BCM rule is applied.

Fig. 3: The training phase in offline learning.

Fig. 4: The testing phase in offline learning.
Summarizing the above results on the sunspot data, 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 is adopted. The state of the ESN is very likely to become divergent due to the fact that the spectral radius is larger than 1. However, the fixed ESN model produces better results in online learning mode than the offline learning mode. Even though the anti-Oja learning rule obtains smallest test error in online learning mode, it does not improve the result in the fixed network.

VI. CONCLUSION

In this work, two plasticity rules have been applied for updating the reservoir weights in ESN model for time series prediction. Our results indicate that the BCM rule is more effective, compared with the anti-Oja rules for enhancing the learning performance in the offline learning mode. In online learning, the anti-Oja rule is more stable than the BCM rule, although neither of them is able to improve the learning performance.

The instability in learning can very likely be attributed to the increased value of spectral radius. However, 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, which hopefully will be more effective and more reliable in enhancing the learning performance of ESNs.

ACKNOWLEDGMENT

This research has supported in part by the Ministry of Higher Education, Malaysia.

REFERENCES