Batch Gradient Training Method with Smoothing l_0 Regularization for Echo State Networks

Zohaib Ahmad(✉), Kaizhe Nie, Junfei Qiao, and Cuili Yang

Faculty of Information Technology, Beijing University of Technology Beijing Key Laboratory of Computational Intelligence and Intelligence System,
Beijing 100124, People's Republic of China
ahmedzohaib03@gmail.com, 1036685809@qq.com

Abstract. The echo state networks (ESNs) have been widely used for time series prediction, due to their excellent learning performance and fast convergence speed. However, the obtained output weight of ESN by pseudoinverse is always ill-posed. In order to solve this problem, the ESN with batch gradient method and smoothing ℓ_0 regularization (ESN-BGSL0) is studied. By introducing a smooth ℓ_0 regularizer into the traditional error function, some redundant output weights of ESN-BGSL0 are driven to zeros and pruned. Two examples are performed to illustrate the efficiency of the proposed algorithm in terms of estimation accuracy and network compactness.

Keywords: Each state networks · Gradient method · ℓ_0 regularization · Sparsity

1 Introduction

Recently, the artificial neural networks are widely used to fit nonlinear dynamic system with arbitrary precision [1]. The typical artificial neural networks include radial basis function neural network (RBF) [2], echo state network (ESN) [3], fuzzy neural network [4], hopfield network [5], and so on. Among these networks, ESN have gained many attentions. As a kind of recursive artificial neural network, the ESN is consisted of an input layer, a reservoir and an output layer [6]. In the training process, only the output weights are trained. Hence, the computational burden of ESN is less than other artificial neural networks.

The performance of an ESN is closely related with its reservoir size. If the reservoir contains too many nodes, the training error may be small, but the over-training problem also exists which leads to high computational complexity and poor generalization performance. If the reservoir size is too small, the ESN has to face the under-training problem. To optimize network size, many algorithms have been proposed, among which the growing and pruning methods are two main trends [7–12]. The growing ESN starts with a small network

X. B. Zhai et al. (Eds.): MLICOM 2019, LNICST 294, pp. 491–500, 2019.
https://doi.org/10.1007/978-3-030-32388-2_42

and adds reservoir nodes one-by-one or group-by-group in the network training process [8,9], this operation requires a long training time which results to heavy computational burden. On the other hand, the pruning method initializes with a large network and removes reservoir neurons or output weights in the learning process [10–12], the network performance can be increased by pruning the unnecessary neurons or weights. Hence, the pruning method is focused in this paper.

To generate sparse network architecture, the regularization methods are studied by adding the norm of weights into the corresponding objective function [13–20]. The commonly used regularization methods include the ℓ_2 regularization [16], ℓ_1 regularization [17,18], $\ell_{1/2}$ regularization [19] and ℓ_0 regularization [20]. As illustrated in [16], the ℓ_2 regularizer is able to control the risk of error amplification, but it is a biased estimation. The ℓ_1 regularization based algorithms could reduce network complexity. However, the ℓ_1 regularizer cannot satisfy the oracle property. Moreover, the $\ell_{1/2}$ regularization has the unbiasedness, sparsity and oracle properties. While, the $\ell_{1/2}$ regularization is not differentiable at the origin which causes oscillations in the training process. Furthermore, according to the regularization theory, the ℓ_0 regularizer is able to yield the most sparse solution among all the regularization based algorithms [20]. However, the ℓ_0 regularization is a NP-hard problem which is difficult to solve.

To optimize network size, the ESN with batch gradient method and smoothing ℓ_0 regularizer (ESN-BGSL0) is proposed in this paper. Since the ℓ_0 regularization penalty term is a NP-hard optimization problem, a continuous function is used to approximate the ℓ_0 regularizer. In ESN-BGSL0, only the output weights are updated by using the batch gradient method and smoothing ℓ_0 regularization, hence its computation complexity is greatly reduced than the traditional recurrent neural networks. Finally, two time series experiments are carried out to show the effectiveness of the proposed algorithm in terms of estimation accuracy and network sparsity.

The rest paper is organized as follows. The original ESN is introduced in Sect. 2. The proposed ESN-BGSL0 is described in Sect. 3. The experiments are done in Sect. 4. Finally, some conclusions are drawn in Sect. 5.

2 Preliminariies

The structure of an original ESN (OESN) without feedback connections is illustrated in Fig. 1. Without loss of generality, it is supposed that the OESN has n input nodes, N neurons and 1 output unit. For given L training samples $\{\mathbf{u}(k), t(k)\}_{k=1}^{L}$, where $\mathbf{u}(k) = [u_1(k), u_2(k), ..., u_n(k)]^T \in \mathbb{R}^n$ are inputs and $t(k)$ denote outputs, the echo states $\mathbf{x}(k) \in \mathbb{R}^N$ at the time step k is calculated as below:

$$\mathbf{x}(k) = \mathbf{g}(\mathbf{W}\mathbf{x}(k-1) + \mathbf{W}^{in}\mathbf{u}(k)) \tag{1}$$

where $\mathbf{g}(\cdot) = [g_1(\cdot), ..., g_N(\cdot)]^T$ are the activation functions of reservoir neurons, $\mathbf{W}^{in} \in \mathbb{R}^{N\times n}$ stands for the input weight and $\mathbf{W} \in \mathbb{R}^{N\times N}$ is the internal weight

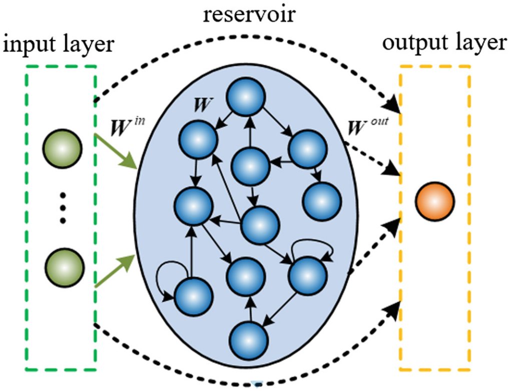


Fig. 1. The structure of the ESN without output feedback.

of reservoir, these two matrices are unchanged after initialization. Then, the OESN output $\mathbf{y}(k)$ at the step k is updated by the following equation:

$$y(k) = \mathbf{W}^{out}(k)\mathbf{x}(k) \tag{2}$$

where $\mathbf{W}^{out} = (W_1, W_2, ..., W_{N+n})^T \in \mathbb{R}^{n+N}$ is the output weight matrix, which is only updated during the learning process.

Now, suppose $\mathbf{X} = [\mathbf{x}(1), \mathbf{x}(2), ..., \mathbf{x}(L)]^T$ represent the internal state matrix and $\mathbf{T} = [\mathbf{t}(1), \mathbf{t}(2), ..., \mathbf{t}(L)]^T$ stand for the target output matrix, the output weights $\mathbf{W}^{out}$ can be computed by minimizing the mean square error as below:

$$\tilde{E}(\mathbf{W}^{out}) = \frac{1}{2}\left\|\mathbf{X}\mathbf{W}^{out} - \mathbf{T}\right\|_2^2 \tag{3}$$

where $\|\cdot\|_2$ denotes the ℓ_2 norm. The solution of $\mathbf{W}^{out}$ is commonly solved by using pseudoinverse [21]:

$$\mathbf{W}^{out} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{T} \tag{4}$$

3 The Proposed ESN-BGSL0

Generally speaking, if the network size N is too large or the training data contains too much noise, the solution of Eq. (4) is likely to be ill-posed, which results in the poor prediction model. To solve this problem, the regularization method is introduced into ESN to prune network output weights and increase its estimation performance. By adding the ℓ_0 regularization item into Eq. (3), the conventional cost function is rewritten as below:

$$E(\mathbf{W}^{out}) = \tilde{E}(\mathbf{W}^{out}) + \lambda\left\|\mathbf{W}^{out}\right\|_0^0 \tag{5}$$

where λ is the regularization coefficient to balance the tradeoff between training accuracy and network compactness, $\left\|\mathbf{W}^{out}\right\|_0^0$ represents the ℓ_0 regularizer [22], which is calculated by $\left\|\mathbf{W}^{out}\right\|_0 = (|W_1|^0 + |W_2|^0 + \cdots + |W_{N+n}|^0)$ for $\mathbf{W}^{out} = (W_1, W_2 \cdots W_{N+n})^T$.

Since the ℓ_0 norm minimization is a NP-hard problem, the following continuous function $f(\cdot)$ is used to approximate the ℓ_0 regularizer,

$$f(\mathbf{W}^{out}) = \sum_{i=1}^{n+N} f(W_i) \tag{6}$$

where $f(W_i)$ is a continuous differentiable function on $\mathbb{R}$ and defined as below

$$f(W_i) = 1 - \frac{\varphi}{{W_i}^2 + \varphi^2} \tag{7}$$

where φ is set to a positive value. Based on Eqs. (6) and (7), the proposed cost function Eq. (5) can be rewritten as below

$$E(\mathbf{W}^{out}) = \tilde{E}(\mathbf{W}^{out}) + \lambda f(\mathbf{W}^{out}) \tag{8}$$

The gradient of the cost function Eq. (8) is given as

$$\frac{\partial E(\mathbf{W}^{out})}{\partial \mathbf{W}^{out}} = -\mathbf{X}^{\mathbf{T}}(\mathbf{T} - \mathbf{X}\mathbf{W}^{out}) + \lambda \frac{\partial f(\mathbf{W}^{out})}{\partial \mathbf{W}^{out}} \tag{9}$$

with

$$\frac{\partial f(\mathbf{W}^{out})}{\partial \mathbf{W}^{out}} = \sum_{i=1}^{n+N} \frac{\partial f(W_i)}{\partial W_i} = \sum_{i=1}^{n+N} \frac{2\varphi W_i}{(W_i + \varphi^2)^2} \tag{10}$$

With an arbitrary initial value, the output weights can be iteratively updated by the batch gradient method,

$$\mathbf{W}^{out}(j+1) = \mathbf{W}^{out}(j) - \eta \frac{\partial E(\mathbf{W}^{out})}{\partial \mathbf{W}^{out}} \tag{11}$$

where $\eta > 0$ is the pre-defined learning rate and j is the updating iteration.

Based on above discussion, the operational process of the proposed ESN-BGSL0 can be summarized as below,

Step 1. Randomly generate an initial reservoir weight matrix $\mathbf{W}_0$ with predefined sparsity and reservoir size N, then update the matrix $\mathbf{W}_0$ as $\mathbf{W}{=}\alpha_{\mathbf{W}}\mathbf{W}_0/\rho(\mathbf{W}_0)$, where $0 < \alpha_{\mathbf{W}} < 1$ and $\rho(\mathbf{W}_0)$ is the spectral radius of $\mathbf{W}_0$. Furthermore, initialize the input weight matrix $\mathbf{W}^{in}$.
Step 2. Drive the reservoir by input signals as shown in Eq. (2), collect the reservoir states to obtain the internal state matrix $\mathbf{X}$.
Step 3. Set $j = 0$, initial the output weights matrix $\mathbf{W}^{out}$.
Step 4. Increase $j = j+1$, With the predefined learning rate η, regularization coefficient λ and positive value φ, update the output weights matrix $\mathbf{W}^{out}(j)$ according to Eqs. (7) to (11).
Step 5. If j reaches to the predefined maximum iteration J, the algorithm stops; Otherwise, turn to Step 4.

4 Simulation Results and Discussion

In this section, the effectiveness of the proposed ESN-BGLS0 is evaluated by the root mean square error (RMSE) [2], which is defined as follows

$$RMSE = \sqrt{\sum_{k=1}^{L} \frac{(t(k) - y(k))^2}{L}} \tag{12}$$

where $t(k)$ and $y(k)$ denote the kth target and ESN outputs, respectively, L is the number of training sample. Moreover, the ESN-BGLS0 is tested on the Lorenz time series prediction and Mackey-Glass time series prediction problems.

4.1 Lorenz Time Series Prediction

As a chaotic dynamical time series, the Lorenz system is governed by the following equations [21]

$$\begin{aligned} \frac{dx}{di} &= a(-x + y) \\ \frac{dy}{di} &= bx - y - xz \\ \frac{dz}{di} &= xy - cz \end{aligned} \tag{13}$$

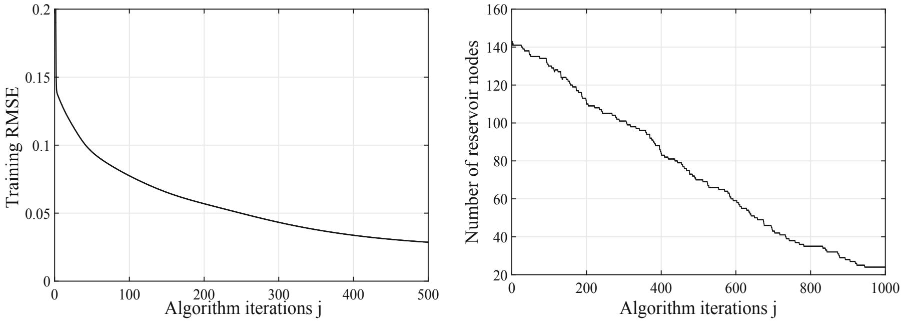
where the parameters are set as $a = 10$, $b = 28$ and $c = 8/3$. The Runge-Kutta method with step 0.01 is used to generate the Lorenz time series values. In each pair of training samples and test samples, $y(k-3)$, $y(k-2)$ and $y(k-1)$ are used to predict $y(k)$. In addition, the initial reservoir size is set as 300. In this experiment, 2400 samples are generated, in which 1200 samples are used as the training dataset and the remaining values are treated as testing dataset.

To study the effectiveness of regularization coefficient λ on network performance, the training RMSE values and the resulted network size $\tilde{N}$ with different λ are illustrated in Table 1. It is noted that the learning rate is set as $\eta = 0.01$ and $\varphi = 0.05$. It is easily found that too large ($\lambda = 10$) or too small ($\lambda = 0$) regularization parameter cannot generate good prediction accuracy. While the proper value $\lambda = 1$ could obtain the sparse network topology $\tilde{N} = 24$ and good training RMSE value 0.0291. Therefore, the determination of regularization parameter is critical for ESN-BGLS0.

The evolving process of training RMSE and the number of non-zero output weights number versus algorithm iterations j are shown in Fig. 2(a) and (b), respectively. It is easily found that when j increases, the training RMSE decreases monotonically and tends to a constant value. Simultaneously, the network size is gradually reduced, which means that the batch gradient with smoothing ℓ_0 regularizer generates the spare network topology.

Table 1. Algorithm parameters for Lorenz time series prediction

λ	10	1	0.1	0.01	0
$\tilde{N}$	4	24	141	147	161
Training RMSE	0.1821	0.0291	0.0304	0.0313	0.0334
Testing RMSE	0.1878	0.0307	0.0327	0.0337	0.0361



(a) The training RMSE values evolving process.

(b) The network size evolving process.

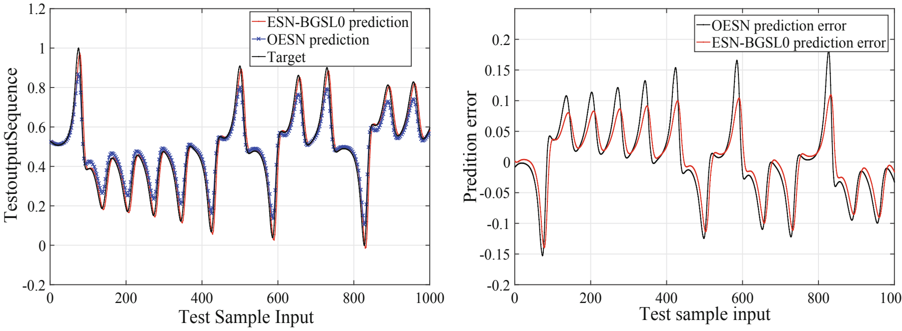
Fig. 2. The evolving process of training RMSE values and network size versus algorithm iterations j.

In the testing phase, the prediction results and testing errors of ESN-BGLS0 and OESN are illustrated in Fig. 3(a) and (b), respectively. It is easily found that the outputs of ESN-BGLS0 could fit to the targets well, also the testing error of ESN-BGLS0 is smaller than that of OESN. This observations imply that the better prediction performance is obtained by ESN-BGLS0 than OESN.

To proof of the effectiveness of ESN-BGLS0, its performance is compared with OESN and the ESN which is trained by the batch gradient (ESN-BG). The comparisons are showed in Table 2, including the training time (s), the training and testing RMSE values, the final network size $\tilde{N}$. From Table 2, it is easily found that the ESN-BGLS0 obtains the smallest training and testing RMSE values with the sparsest network topology among all the evaluated algorithms.

Table 2. Algorithm comparisons for Lorenz time series prediction

Approaches	$\tilde{N}$	Training time(s)	Training RMSE	Testing RMSE
ESN-BGSL0 ($\lambda = 1$)	3	157.217	0.0291	0.0307
ESN-BGSL0	161	150.494	0.0334	0.0361
OESN	300	1.257	0.0519	0.0554



(a) The prediction curves.

(b) The testing error.

Fig. 3. The prediction and testing error of ESN-BGSL0 and OESN.

4.2 Mackey-Glass Time Series Prediction

The Mackey-Glass time series is derived by a time-delay differential system with the following form [23]

$$\frac{\mathrm{d}x}{dt} = \beta x(t) + \frac{ax(t-\delta)}{1 + x(t-\delta)^{10}} \tag{14}$$

where the parameters are set as $\beta = 0$, $\alpha = 0.2$, and $\delta = 17$. The dataset is constructed by the second-order Runge-Kutta method with step size 0.1. In this experiment, 2400 samples are used, in which 1200 samples are used as training dataset and the remaining 1200 values are treated as test dataset.

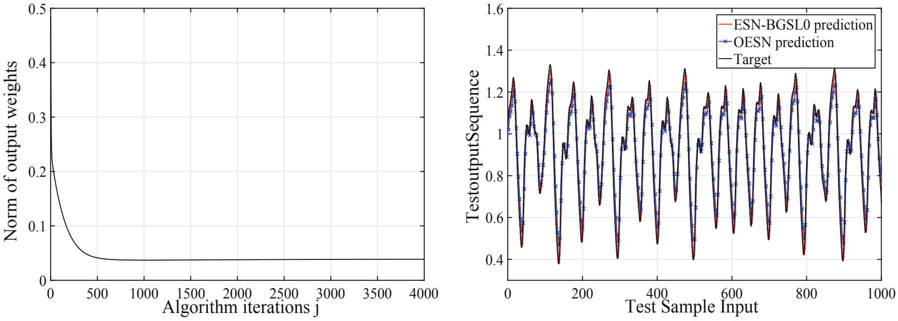
The training RMSE values with different regularization parameters λ are listed in Table 3. Obviously, the too large or too small λ cannot generate good network compactness and training accuracy. To further study the effectiveness of the proposed ESN-BGSL0, the training RMSE values and the number of non-zero output weights versus batch gradient iterations j are shown in Fig. 4(a) and (b), respectively. It can be clearly seen that the training RMSE values is gradually reduced when j is increased. In addition, the complexity of the network is greatly simplified in the training process, which implies the network sparsity is improved.

To evaluate the estimation performance of ESN-BGSL0, the prediction outputs and prediction errors of ESN-BGSL0 and OESN are shown in Fig. 5(a) and (b), respectively. Obviously, the ESN-BGSL0 has smaller prediction error than that of OESN, thus the validity of ESN-BGSL0 is illustrated.

The performance comparisons between different algorithms are given in Table 4. It can be seen that the ESN-BGSL0 has the smallest training RMSE value (0.0346) and the best network size (18), this fact implies that the network estimation accuracy and network compactness have been greatly improved by using the smoothing ℓ_0 regularization penalty term.

Table 3. Algorithm parameters for Mackey-Glass time-series prediction

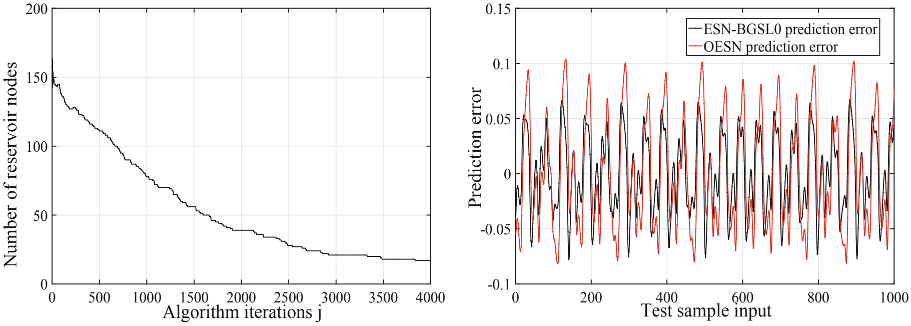
λ	3	0.3	0.03	0.003	0
$\tilde{N}$	4	18	133	139	148
Training RMSE	0.0463	0.0346	0.0350	0.0371	0.0383
Testing RMSE	0.0481	0.0349	0.0354	0.0384	0.0395



(a) The training RMSE values evolving process.

(b) The network size evolving process.

Fig. 4. The evolving process of training RMSE and network size versus algorithm iterations j.



(a) The prediction curves.

(b) The testing error.

Fig. 5. The prediction and testing error of ESN-BGSL0 and OESN for Mackey-Glass time series prediction.

Table 4. Algorithm comparisons for Mackey-Glass time series prediction

Approaches	$\tilde{N}$	Training time(s)	Training RMSE	Testing RMSE
ESN-BGSL0 ($\lambda = 0.3$)	18	627.172	0.0346	0.0349
ESN-BGSL0	148	614.975	0.0383	0.0395
OESN	300	1.168	0.0503	0.0524

5 Conclusions

To solve the ill-posed problem is ESN, the batch gradient method and ℓ_0 regularization are combined together to train and prune ESN topology. In the proposed algorithm, the ℓ_0 norm of output weights are added into the objective function, which is solved by the batch gradient descent algorithm. As illustrated by the simulation results, the proposed ESN owns smaller network size and better prediction accuracy than OESN.

Acknowledgement. This work was supported by the National Natural Science Foundation of China under Grants 61603012, 61533002 and 61890930-5, the Beijing Municipal Education Commission Foundation under Grant KM201710005025, the Major Science and Technology Program for Water Pollution Control and Treatment of China (2018ZX07111005), the National Key Research and Development Project under Grants 2018YFC1900800-5.

References

1. Schäfer, A.M., Zimmermann, H.G.: Recurrent neural networks are universal approximators. In: Kollias, S.D., Stafylopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006. LNCS, vol. 4131, pp. 632–640. Springer, Heidelberg (2006). https://doi.org/10.1007/11840817_66
2. Han, H.G., Chen, Q.L., Qiao, J.F.: An efficient self-organizing RBF neural network for water quality prediction. Neural Netw. Off. J. Int. Neural Netw. Soc. **24**(7), 717–725 (2011)
3. Jaeger, H.: Echo state network. Scholarpedia **2**(9), 1479–1482 (2007)
4. Hayashi, Y., Buckley, J.J., Czogala, E.: Fuzzy neural network with fuzzy signals and weights. Int. J. Intell. Syst. **8**(4), 527–537 (2010)
5. Song, B., Zhang, Y., Shu, Z., et al.: Stability analysis of Hopfield neural networks perturbed by Poisson noises. Neurocomputing **196**(C), 53–58 (2016)
6. Xue, Y., Yang, L., Haykin, S.: Decoupled echo state networks with lateral inhibition. Neural Netw. Off. J. Int. Neural Netw. Soc. **20**(3), 365–376 (2007)
7. Augasta, M.G., Kathirvalavakumar, T.: A novel pruning algorithm for optimizing feedforward neural network of classification problems. Neural Process. Lett. **34**(3), 241–258 (2011)
8. Fan-jun, L., Ying, L.: An approach to design growing echo state networks. In: Yin, H., et al. (eds.) IDEAL 2016. LNCS, vol. 9937, pp. 220–230. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46257-8_24
9. Qiao, J., Li, F., Han, H., et al.: Growing echo-state network with multiple subreservoirs. IEEE Trans. Neural Netw. Learn. Syst. **28**(2), 391–404 (2017)
10. Scardapane, S., Nocco, G., Comminiello, D., et al.: An effective criterion for pruning reservoir's connections in echo state networks. In: International Joint Conference on Neural Networks. IEEE (2015)
11. Scardapane, S., Comminiello, D., Scarpiniti, M., Uncini, A.: Significance-based pruning for reservoir's neurons in echo state networks. In: Bassis, S., Esposito, A., Morabito, F.C. (eds.) Advances in Neural Networks: Computational and Theoretical Issues. SIST, vol. 37, pp. 31–38. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18164-6_4

12. Li, D., Liu, F., Qiao, J., et al.: Structure optimization for echo state network based on contribution. Tsinghua Sci. Technol. **24**(01), 99–107 (2019)
13. Loone, S., Irwin, G.: Improving neural network training solutions using regularisation. Neurocomputing, **37**, 71–90
14. Setiono, R.: A penalty-function approach for pruning feedforward neural networks. Neural Comput. **9**, 185–204
15. Shao, H.M., Xu, D.P., Zheng, G.F., Liu, L.J.: Convergence of an online gradient method with inner-product penalty and adaptive momentum. Neurocomputing, **77**, 243–252
16. Dutoit, X., Schrauwen, B., Campenhout, J.V., et al.: Pruning and regularization in reservoir computing. Neurocomputing **72**(7–9), 1534–1546 (2009)
17. Han, M., Ren, W.-J., Xu, M.-L.: An improved echo state network via L1-norm regularization. Acta Automatica Sinica **40**(11), 2428–2435 (2014)
18. Scardapane, S., Panella, M., Comminiello, D., Hussian, A., Uncini, A.: Distributed reservoir computing with sparse readouts research frontier. IEEE Comput. **11**(4), 59–70 (2016)
19. Wu, W., Fan, Q., Zurada, J.M., et al.: Batch gradient method with smoothing $l_1/2$ regularization for training of feedforward neural networks. Neural Netw. Off. J. Int. Neural Netw. Soc. **50**(2), 72 (2013)
20. Louizos, C., Welling, M., Kingma, D.P.: Learning sparse neural networks through l_0 regularization (2018)
21. Yang, C., Qiao, J., Han, H., et al.: Design of polynomial echo state networks for time series prediction. Neurocomputing **290**, 148–160 (2018)
22. Zhang, H., Tang, Y., Liu, X.: Batch gradient training method with smoothing l_0 regularization for feedforward neural networks. Neural Comput. Appl. **26**(2), 383–390 (2015)
23. Yang, C., Qiao, J., Wang, L., et al.: Dynamical regularized echo state network for time series prediction. Neural Comput. Appl. (3–4), 1–14 (2018)