2022 International Conference on Emerging Techniques in Computational Intelligence (ICETCI) | 978-1-6654-7941-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICETCI55171.2022.9921370

Feature Extraction Mechanism for Each Layer of Deep Echo State Network

Keikou Kanda
Department of Computer Science
Chiba Institute of Technology
Narashino, Japan
s1831047GE@s.chibakoudai.jp

Sou Nobukawa
Department of Computer Science
Chiba Institute of Technology
Narashino, Japan
nobukawa@cs.it-chiba.ac.jp

***Abstract*—The echo state network (ESN) is an efficient machine learning model that is the most typical type of reservoir-computing framework. Recently, research has been conducted on deep echo state networks (deepESN). The deepESN consists of an input layer, multiple reservoir layers, and an output layer, which achieve a very high memory capacity (MC). Furthermore, it has been suggested that deepESN can represent various temporal scales using layer hierarchization. However, the exact role of each layer in the feature extraction has not yet been revealed. Therefore, deepESN parameter adjustments must be conducted using empirical measurements or grid searches based on a trial-and-error method. To establish a design framework for deepESN, revealing the deepESN parameters related to feature extraction is crucial. To analyze the dynamics of neural networks, we applied multiscale entropy (MSE) analysis to a physiological neural network model and found that complex topological features and multiple neural module structures produce complex temporal-scale dependencies. Therefore, we hypothesized that the feature extraction function of each layer in the deepESN could be revealed using MSE analysis. To validate this hypothesis, we analyzed the output of each layer using MSE analysis and MC task. As a result, in this study, under small inner layer connections, a high memory capacity was achieved by temporal scale-specific feature extraction in each layer compared to larger inter-layer connections. In conclusion, this study revealed the feature extraction function in deepESN and provided the parameter setting method, especially regarding interlayer connection as a part of the design framework of deepESN.**

***Index Terms*—Complexity analysis, deep echo state network, multiscale entropy analysis, memory capacity**

I. Introduction

Reservoir computing (RC) is attracting attention as a computational model that realizes high-speed processing of recurrent neural networks (RNN). RC consists of an input layer, reservoir layer, and output layer [1]–[3]. The neural network used in RC can be trained only by adjusting the synaptic weights in the readout part of the output layer, which is particularly advantageous for edge hardware implementations that perform time-series processing [4], [5]. An echo state network (ESN) is one of the most widely used RC frameworks [6]. As a property of ESN, it significantly reduces the synaptic weight adjustments required for their training, which makes learning more efficient compared to long short-term memory (LSTM) as the other mainstream RNN methods through backpropagation over time [5], [7]. Although ESN exhibits high learning efficiency, its learning accuracy is inferior to that of the LSTM approach [5], [8]. Therefore, recent approaches to ESN have focused on the decay factor of the composed neurons, network topology, and architectures based on sets of ESN [9]–[12]. To enhance the decay factor of neurons, a model using chaotic neurons as the neuron model for the component of the reservoir has been proposed [11]. In a study that focused on network topology, small-world network topology also realized a wide parameter region of the echo state property (ESP) in ESN [12]–[16]. Among these studies, deep echo state networks (deepESN) achieve a very high memory capacity [17]. A deepESN is a hierarchical system with multiple reservoir layers, which leads to a complex temporal response with rich temporal-scale dynamics [17]–[20]. However, deepESN parameter adjustments are achieved through empirical measurements or grid searches based on a trial-and-error method. To establish a design framework for deepESN, revealing the deepESN parameters related to feature extraction(i.e., adjusting the strength of connections between layers, neurons, or input signals) is crucial. Furthermore, it has been suggested that deepESN can represent various temporal scales of layers, but the concrete role of each layer in feature extraction has not been revealed [17].

To analyze the properties of a network, we applied multi-scale entropy (MSE) analysis to a physiological neural network model and found that complex topological features and multiple neural module structures produce complex temporal-scale dependencies [21]–[23]. Therefore, we hypothesized that the feature extraction function of each layer in the deepESN could be revealed using MSE analysis. To validate this hypothesis, we analyzed the output of each layer using MSE analysis and Memory Capacity (MC) task [24].

II. Materials and Methods

A. Echo State Network

ESN architecture is shown in Fig.1. The input signal $\mathbf{u}(t) \in \mathbb{R}^{N_U}$ is an N_U-dimensional input signal that is applied to N_R neurons in the reservoir network. The dynamics of the firing state $\mathbf{x}(t) \in \mathbb{R}^{N_R}$ driven by the input signal $\mathbf{u}(t)$ and recurrent signals are given by

$$\mathbf{x}(t) = \tanh(W_{in}\mathbf{u}(t) + \boldsymbol{\theta} + W\mathbf{x}(t-1)), \qquad (1)$$

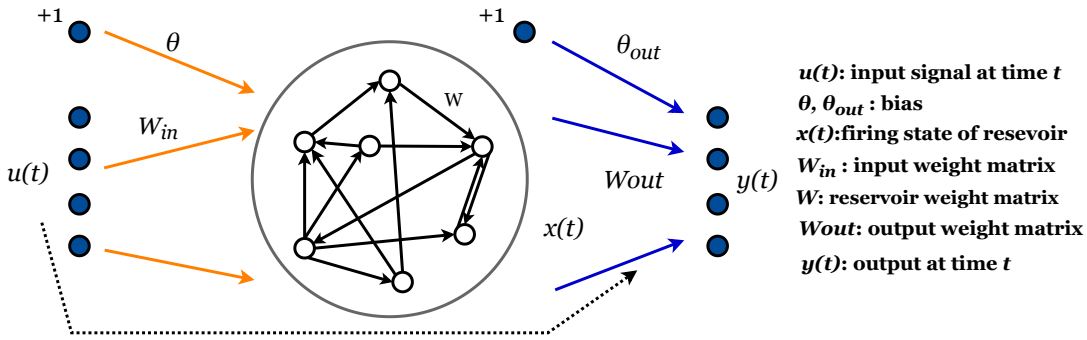

Fig. 1. The architecture of a ESN model : $\mathbf{u}(t) \in \mathbb{R}^{N_U}$ and $\mathbf{x}(t) \in \mathbb{R}^{N_R}$ represent the input and the resevoir state at time t, $W_{in} \in \mathbb{R}^{N_R \times N_R}$ is the input-to-reservoir layer weight matrix, $\boldsymbol{\theta} \in R^{N_R}$ is the bias-to-reservoir weight vector (where we assume that the input bias is equal to 1 for the reservoir unit), $W \in \mathbb{R}^{N_R \times N_R}$ current reservoir weight matrix

where $W_{in} \in \mathbb{R}^{N_R \times N_R}$ denotes the input-to-reservoir layer weight matrix, $\boldsymbol{\theta} \in \mathbb{R}^{N_R}$ denotes the bias-to-reservoir weight vector, $\tanh$ denotes the hyperbolic tangent activation function applied per element, and $W \in \mathbb{R}^{N_R \times N_R}$ denotes the reservoir weight matrix. The values of matrix W are random matrices with spectral radius α, according to the following process: First, random matrix W_0 is produced from a uniform distribution. Then, based on the spectral radius W_0: $\rho(W_0)$, the spectral radius is rescaled to α [25]:

$$W = \alpha \frac{W_0}{\rho(W_0)}. \tag{2}$$

Accordingly, the weight values in W_{in} and $\boldsymbol{\theta}$ are chosen from the uniform distribution.

B. Deep Echo State Network

This section summarizes the dynamics of the deepESN used in this research. This study focused on a straight stack of reservoirs, which showed the highest performance in a previous study [17]. The model that we consider is a straight stack of reservoirs, called deepESN and shown in Fig.2. The deepESNs dynamics of firing state $\mathbf{x}^{(l)}(t)$ can be expressed as:

$$\mathbf{x}^{(l)}(t) = \tanh(W_{in}^{(l)}\mathbf{I}(t) + \boldsymbol{\theta}^{(l)} + W^{(l)}\mathbf{x}^{(l)}(t-1)), \tag{3}$$

where the superscript (l) is used to refer to the network parameters and hyperparameters at layer l. For the sake of simplicity, the same number of reservoir units N_R is present in each layer of the stack, $\boldsymbol{\theta}^{(l)} \in \mathbb{R}^{N^R}$ is the bias-to-reservoir weight vector for layer l, $W^{(l)} \in \mathbb{R}^{N^R \times N^R}$ represents the recurrent weight matrix of layer l, and $W_{in}^{(l)}$ denotes the input weight matrix for layer l. $W_{in}^{(l)}$ are chosen from the uniform distribution within the range $[-scale_{in}, scale_{in}]$, where $scale_{in}$ is called an input-scaling parameter for each layer. Moreover, $\mathbf{I}^{(l)}(t)$ in Eq.(3) is used to denote the input for the $l-$th layer of the deeppESN architecture at time step t, that is,

$$\mathbf{I}^{(l)}(t) = \begin{cases} \mathbf{u}(t) & \text{if } \ l = 1 \\ \mathbf{x}^{(l-1)}(t) & \text{if } \ l > 1. \end{cases} \tag{4}$$

For the output calculation in a deepESN, the outputs of all reservoir units were linearly combined, as in a standard ESN. Considering the hierarchical structure of reservoirs and using the number of reservoir layers N_L, the output of DeepESN at each time step t can be computed as

$$\mathbf{y}(t) = W_{out}[\mathbf{x}^{(1)}(t)\mathbf{x}^{(2)}(t)...\mathbf{x}^{(N_L)}(t)] + \boldsymbol{\theta}_{out}, \tag{5}$$

where $\mathbf{y}(t) \in \mathbb{R}^{N_Y}$ is the N_Y-dimensional output at time t, and $W_{out} \in \mathbb{R}^{N_Y \times N_L N_R}$ represents the reservoir-to-readout weight matrix of the deepESN that connects the reservoir units in all layers to the units in the readout. Therefore, training a deepESN can be accomplished through direct methods similar to the case of a standard ESN [17].

C. Evaluation index

1) Memory Capacity: The approach proposed for the MC task can be used to evaluate the performance of the reservoir [24]. This task can provide a measure of the short-term memory capacity in reservoirs. The MC is defined as the coefficient of determination of the input and output before the τ_m step. where the input signal $\mathbf{u}(t)$ is a random value extracted from a uniform distribution of $[-0.8, 0.8]$ and the teacher signal $y_d(t)$ is given by $u(t - \tau_m)$. MC is calculated by:

$$MC = \sum_{\tau_m=1}^{T} MC_{\tau_m}. \tag{6}$$

$$MC_{\tau_m} = \frac{\mathrm{cov}^2(u_{\tau_m}, y)}{\sigma^2(u_{\tau_m})\sigma^2(y)}. \tag{7}$$

$\mathrm{cov}(u_{\tau_m}, y)$ is the covariance between the teacher signal and output of ESN. $\sigma^2(u_{\tau_m})$ and $\sigma^2(y)$ represent the variances in the teacher signal and output, respectively. In this study, the input signal contained 6,000 time-series data, of which 5,000 steps were used for training, and the remaining 1,000 steps were used as teacher data. The setup of the MC task is based on prior work on various parameters [26]. In particular, we implemented a deepESN architecture with $N_R = 10$ fully connected reservoir units, $N_L = 10$ reservoir layers, scaling of the input signal, and $\rho = 0.9$ spectral radius between layers.

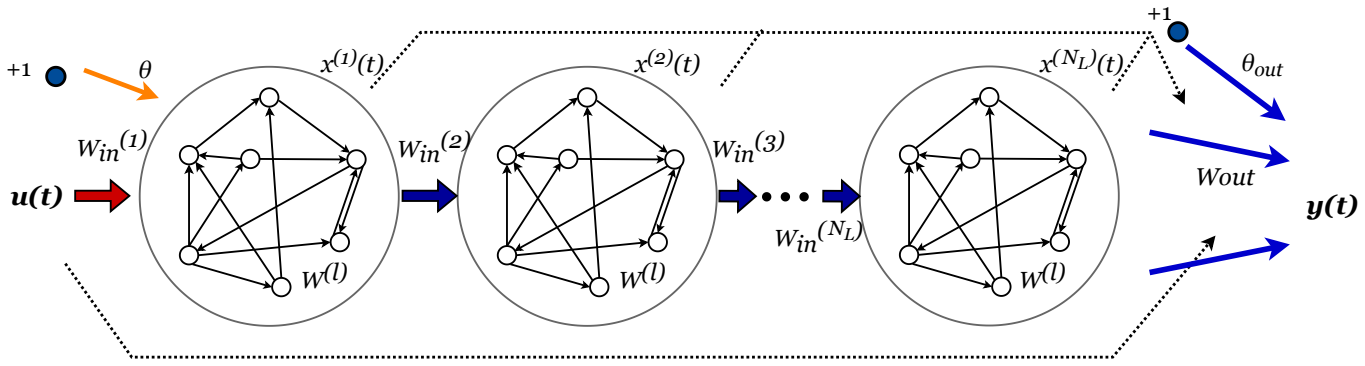

Fig. 2. The architecture of a deepESN model : $\mathbf{u}(t) \in \mathbb{R}^{N_U}$ and $\mathbf{x}(t) \in \mathbb{R}^{N_R}$ represent the input and the resevoir state of each layer at time t, $W_{in}^{(l)}$ denotes the input weight matrix for layer l, $\boldsymbol{\theta}^{(l)} \in \mathbb{R}^{N^R}$ is the bias-to-reservoir weight vector for layer l and $W^{(l)} \in \mathbb{R}^{N^R \times N^R}$ represents the recurrent weight matrix of layer l.

2) Multiscale Entropy analysis: MSE analysis is a method of quantifying the complexity of time series data at multiple temporal scales by coarse-grading [23].

First, the output time series for each layer $\mathbf{x}^{(l)}(t)$ was coarse-grained using the temporal scale factor (τ) with a non-overlapping window, and subsequently z-scored: $\{z_i, z_{i+1}, ..., z_{i+m-1}\}$. The coarse-grained time series with $\tau = 1$ was identical to the original time series, and a larger τ represented a longer temporal scale. The complexity of the coarse-grained time series output was quantified at each layer using sample entropy (SampEn). SampEn is the natural logarithm of the number of $(m+1)$ consecutive datasets in dataset (t) following m consecutive datasets that are similar within the allowed range (r). SampEn can be computed as

$$S_E(m, r) = -\log \frac{U_{m+1}(r)}{U_m(r)}. \tag{8}$$

$U_m(r)$ is the probability that $|z_i^m - z_j^m| < r (i \neq j, i, j = 1, 2, ...)$. z_j^m is an m-dimensional vector, $z_i^m = \{z_i, z_{i+1}, ..., z_{i+m-1}\}$. $\{z_i, z_{i+1}, ..., z_{i+m-1}\}$ is the coarse-grained output time series for each layer in the deepESN. $\tau(\tau = 1, 2, ...)$ is the temporal scale. In this study, we set $m = 2$ and $r = 0.2$ [23].

III. Results

A. Dependence in Memory Capacity on Input Signal Scaling

First, we investigate the influence of setting scaling value $scale_{in}$ to MC. Figure 3 shows dependence of MC on τ_m at different $scale_{in}$ values ($scale_{in} = 0.1, 1$) As a result, the MC value in the $scale_{in} = 0.1$ maintained MC ≈ 1.0 in $\tau_m < 40$. For larger τ_m, the MC value decreases and subsequently converges to $\tau_m = 100$. On the other hand, in the case of $scale_{in} = 1.0$, MC ≈ 1.0 is maintained at $\tau_m < 8$; MC converges to 0 at $\tau_m = 11$. Therefore, the deepESN with $scale_{in} = 0.1$ holds memory for a longer duration than that with $scale_{in} = 1.0$.

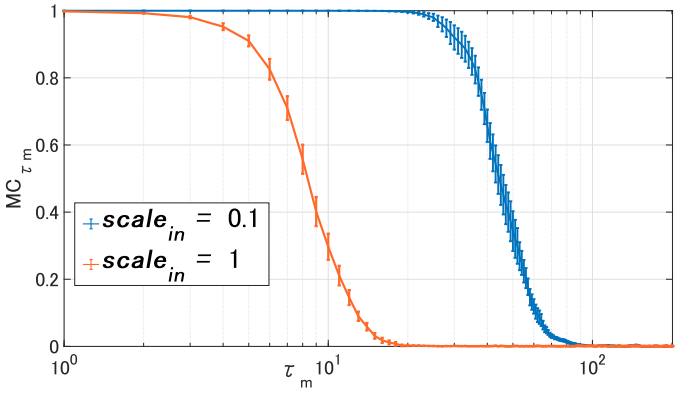

Fig. 3. Memory capacity (MC) performance of deepESN as function of τ_m under the condition for input scaling settings $scale_{in} = 0.1$ (blue line), 1 (orange line). Solid line and error bars indicate the mean and standard deviation of ten trials. The deepENS with $scale_{in} = 0.1$ had a higher memory capacity than that with $scale_{in} = 1.0$, which holds memory up to $\tau_m = 100$.

B. Dynamics of firing state in each layer

To assess the function of feature extraction in each layer, the dynamics of the firing state $\mathbf{x}^{(l)}(t)$ are evaluated. Figure 4 shows the averaged time-series $\mathbf{x}^{(l)}(t)$ among the neurons within each layer: $\bar{\mathbf{x}}^{(l)}(t)$ for $l = 1-10$ for $scale_{in} = 0.1$ (part (a)), 1 (part (b)). In $scale_{in} = 0.1$, layer-specific temporal-scale characteristics were confirmed, for example, irregular slow temporal behavior in $l = 2, 3, 4$ cases and fast periodic oscillations in $l = 6, 8, 9, 10$. However, in the $scale_{in} = 1.0$ case, such layer-specific temporal scale characteristics were not confirmed.

Furthermore, to quantify the layer-specific temporal-scale characteristics, MSE analysis was applied to $\bar{x}^{(l)}(t)$ (see Fig.5). At $scale_{in} = 0.1$, among all scale factors τ, significantly different values of SampEn among layers were confirmed. In contrast, in $scale_{in} = 1.0$, the degree of difference in SampEn among the layers is relatively small in comparison with $scale_{in} = 0.1$.

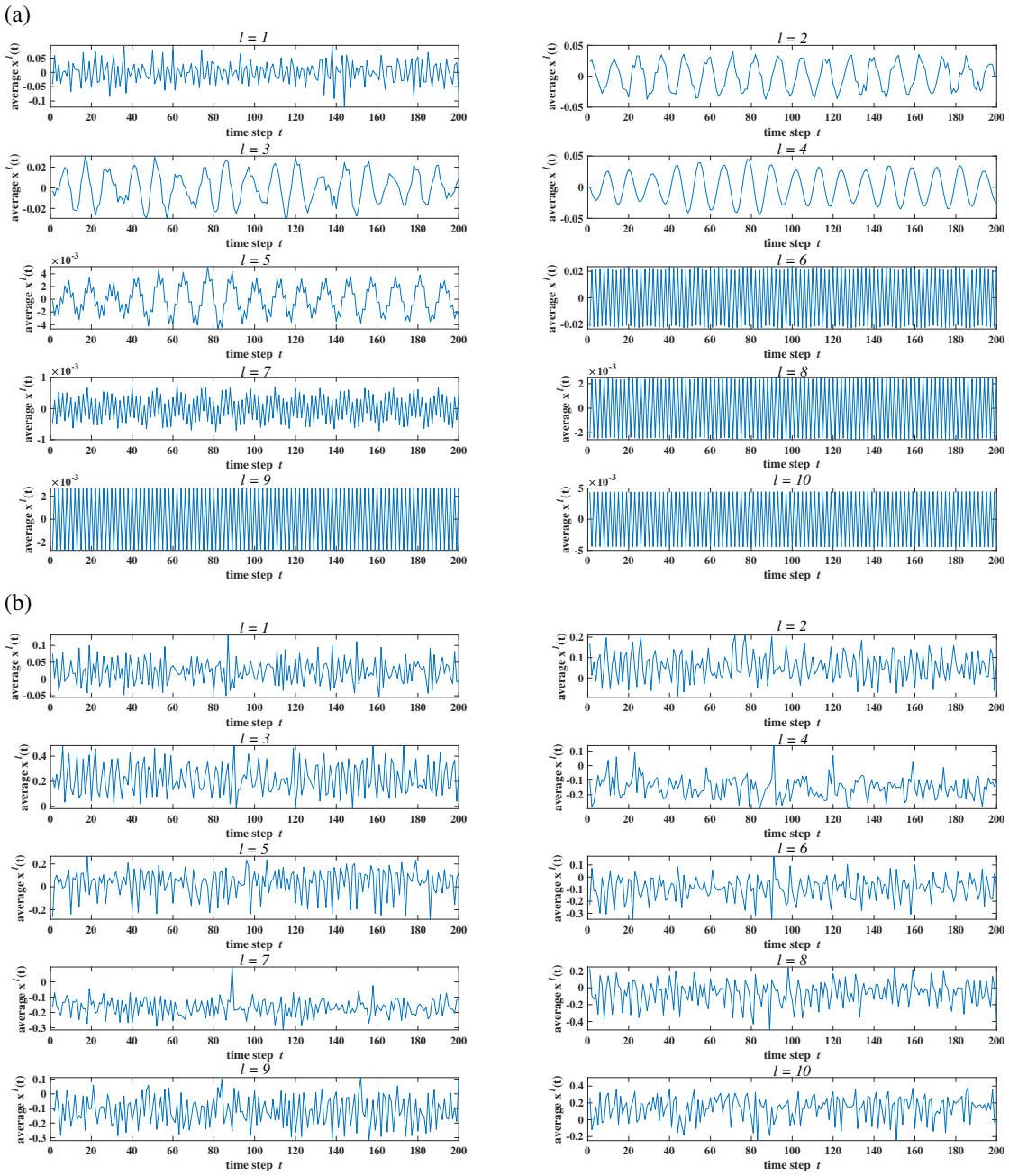

Fig. 4. Averaged time-series $\mathbf{x}^{(l)}(t)$ among neurons within each layer: $\bar{\mathbf{x}}^{(l)}(t)$ at each layer $l = 1 - 10$ (a) $scale_{in} = 0.1$ case. (b) $scale_{in} = 1$. In $scale_{in} = 0.1$, layer-specific temporal-scale characteristics were confirmed, for example, irregular slow temporal behavior in $l = 2, 3, 4$ cases and fast periodic oscillations in $l = 6, 8, 9, 10$. While, in the $scale_{in} = 1.0$ case, such layer-specific temporal scale characteristics were not confirmed.

IV. Discussion and Conclusions

In this study, we evaluated the performance of DeepESN and the dynamics of the firing state in each layer. As a result, under the small $W_{in}^{(l)}$ condition (corresponding to $scale_{in} = 0.1$), a high memory capacity was achieved by virtue of feature extraction for the temporal scale specific in each layer.

We must consider why layer-specific dynamical patterns appear (see Figs.4 and 5) under the small $W_{in}^{(l)}$ condition. The amplitude of the input signal to each layer with a small $W_{in}^{(l)}$ ($scale_{in} = 0.1$) becomes smaller than that in the case of a large $W_{in}^{(l)}$ condition ($scale_{in} = 1.0$). Therefore, in the deeper layer, mutual interactions among neurons within the layer are altered by decreasing the number of firing neurons. Consequently, the complexity of the output significantly is different among layers in the case with a small $W_{in}^{(l)}$ condition. This phenomenon might contribute to the achievement of highly effective feature extraction and a high MC (see Fig.3). However, for a large $W_{in}^{(l)}$ condition ($scale_{in} = 1.0$), the

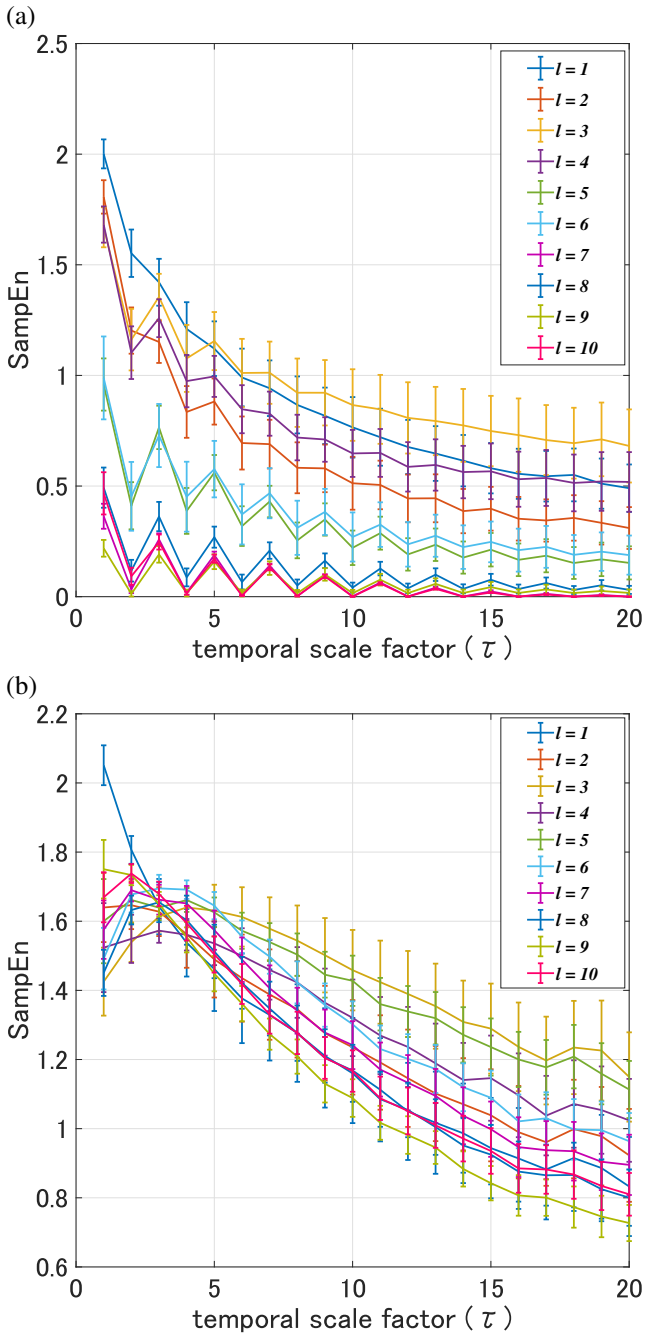

Fig. 5. Dependence of sample entropy (SampEn) for averaged temporal scales $\mathbf{x}^{(l)}(t)$ at each output layer. (a) $scale_{in} = 0.1$. (b) $scale_{in} = 1.0$. The solid line and error bars show the mean and standard error of the trials. At $scale_{in} = 0.1$, among all scale factors τ, significantly different values of SampEn among layers were confirmed. In contrast, in $scale_{in} = 1.0$, the degree of difference in SampEn among the layers is relatively small in comparison with $scale_{in} = 0.1$.

depression of the amplitude of the input signal does not occur; therefore, this mechanism for feature extraction cannot be achieved.

Finally, we must consider the limitation of this study. The decay factor induced by feedback affects MC. However, this functional feedback pathway in each $scale_{in}$ has not been conducted in this study. Therefore, in our future works, we will deal with this point.

In conclusion, this study revealed the feature extraction function in deepESN and provided the parameter setting method, especially regarding the inter-layer connection typified as $W_{in}^{(l)}$, as a part of the design framework of deepESN. In future work, the functions of each layer must be evaluated against not only the MC task but also various prediction/classification tasks.

ACKNOWLEDGMENT

This study was supported by JSPS KAKENHI for a Grant-in-Aid for Scientific Research (C) (Grant No. 22K12183) (SN).

REFERENCES

[1] Herbert Jaeger. The “echo state” approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, Vol. 148, No. 34, p. 13, 2001.

[2] Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, Vol. 3, No. 3, pp. 127–149, 2009.

[3] Sou Nobukawa, Haruhiko Nishimura, and Teruya Yamanishi. Pattern classification by spiking neural networks combining self-organized and reward-related spike-timing-dependent plasticity. *Journal of Artificial Intelligence and Soft Computing Research*, Vol. 9, No. 4, pp. 283–291, 2019.

[4] Kohei Nakajima and Ingo Fischer. Reservoir computing. *Springer*, Vol. 1, No. 5, p. 8, 2021.

[5] Gouhei Tanaka, Toshiyuki Yamane, Jean Benoit Héroux, Ryosho Nakane, Naoki Kanazawa, Seiji Takeda, Hidetoshi Numata, Daiju Nakano, and Akira Hirose. Recent advances in physical reservoir computing: A review. *Neural Networks*, Vol. 115, pp. 100–123, 2019.

[6] Herbert Jaeger. Echo state network. *scholarpedia*, Vol. 2, No. 9, p. 2330, 2007.

[7] Hojjat Salehinejad, Sharan Sankar, Joseph Barfett, Errol Colak, and Shahrokh Valaee. Recent advances in recurrent neural networks. *arXiv preprint arXiv:1801.01078*, 2017.

[8] Claudio Gallicchio, Alessio Micheli, and Luca Pedrelli. Comparison between deepesns and gated rnns on multivariate time-series prediction. *arXiv preprint arXiv:1812.11527*, 2018.

[9] Qianli Ma, Enhuan Chen, Zhenxi Lin, Jiangyue Yan, Zhiwen Yu, and Wing WY Ng. Convolutional multitimescale echo state network. *IEEE Transactions on Cybernetics*, Vol. 51, No. 3, pp. 1613–1625, 2019.

[10] Shisheng Zhong, Xiaolong Xie, Lin Lin, and Fang Wang. Genetic algorithm optimized double-reservoir echo state network for multi-regime time series prediction. *Neurocomputing*, Vol. 238, pp. 191–204, 2017.

[11] Yudai Ebato, Sou Nobukawa, and Haruhiko Nishimura. Effect of neural decay factors on prediction performance in chaotic echo state networks. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1888–1893. IEEE, 2021.

[12] Yuji Kawai, Jihoon Park, and Minoru Asada. A small-world topology enhances the echo state property and signal propagation in reservoir computing. *Neural Networks*, Vol. 112, pp. 15–23, 2019.

[13] Hongyan Cui, Xiang Liu, and Lixiang Li. The architecture of dynamic reservoir in the echo state network. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, Vol. 22, No. 3, p. 033127, 2012.

[14] Fenglan Chen, Yongjun Shen, Guidong Zhang, and Xin Liu. The network security situation predicting technology based on the small-world echo state network. In *2013 IEEE 4th International Conference on Software Engineering and Service Science*, pp. 377–380. IEEE, 2013.

[15] Xugang Xi, Wenjun Jiang, Seyed M Miran, Xian Hua, Yun-Bo Zhao, Chen Yang, and Zhizeng Luo. Feature extraction of surface electromyography based on improved small-world leaky echo state network. *Neural Computation*, Vol. 32, No. 4, pp. 741–758, 2020.

[16] Shiping Wen, Rui Hu, Yin Yang, Tingwen Huang, Zhigang Zeng, and Yong-Duan Song. Memristor-based echo state network with online least mean square. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 49, No. 9, pp. 1787–1796, 2018.

[17] Claudio Gallicchio, Alessio Micheli, and Luca Pedrelli. Deep reservoir computing: A critical experimental analysis. *Neurocomputing*, Vol. 268, pp. 87–99, 2017.

[18] Zeeshan Khawar Malik, Amir Hussain, and Qingming Jonathan Wu. Multilayered echo state machine: A novel architecture and algorithm. *IEEE Transactions on cybernetics*, Vol. 47, No. 4, pp. 946–959, 2016.

[19] Taha Ait Tchakoucht and Mostafa Ezziyyani. Multilayered echo-state machine: a novel architecture for efficient intrusion detection. *IEEE Access*, Vol. 6, pp. 72458–72468, 2018.

[20] Jianyu Long, Shaohui Zhang, and Chuan Li. Evolving deep echo state networks for intelligent fault diagnosis. *IEEE Transactions on Industrial Informatics*, Vol. 16, No. 7, pp. 4928–4937, 2019.

[21] Sou Nobukawa, Haruhiko Nishimura, and Teruya Yamanishi. Temporal-specific complexity of spiking patterns in spontaneous activity induced by a dual complex network structure. *Scientific reports*, Vol. 9, No. 1, pp. 1–12, 2019.

[22] Sou Nobukawa, Haruhiko Nishimura, Nobuhiko Wagatsuma, Satoshi Ando, and Teruya Yamanishi. Long-tailed characteristic of spiking pattern alternation induced by log-normal excitatory synaptic distribution. *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 32, No. 8, pp. 3525–3537, 2020.

[23] Madalena Costa, Ary L Goldberger, and C-K Peng. Multiscale entropy analysis of biological signals. *Physical review E*, Vol. 71, No. 2, p. 021906, 2005.

[24] Herbert Jaeger, et al. *Short term memory in echo state networks*, Vol. 5. GMD-Forschungszentrum Informationstechnik, 2001.

[25] Yuji Kawai, Jihoon Park, and Minoru Asada. A small-world topology enhances the echo state property and signal propagation in reservoir computing. *Neural Networks*, Vol. 112, pp. 15–23, 2019.

[26] Benjamin Schrauwen, Marion Wardermann, David Verstraeten, Jochen J Steil, and Dirk Stroobandt. Improving reservoirs using intrinsic plasticity. *Neurocomputing*, Vol. 71, No. 7-9, pp. 1159–1171, 2008.