2023 International Symposium on Nonlinear Theory and Its Applications
NOLTA2023, September 26-29, 2023, Catania and Online

Ensemble learning of multiple readouts for reservoir computing

Yuichiro Tanaka† and Hakaru Tamukoh†‡

†Research Center for Neuromorphic AI Hardware, Kyushu Institute of Technology
2–4 Hibikino, Wakamatsu, Kitakyushu, 808-0196, Japan
‡Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology
2–4 Hibikino, Wakamatsu, Kitakyushu, 808-0196, Japan

Email: tanaka-yuichiro@brain.kyutech.ac.jp, tamukoh@brain.kyutech.ac.jp

Abstract— This study proposes a reservoir computing model with multiple readouts and an associated training method to enhance the training capability of reservoir computing. This study conducts a speaker classification task and a word classification task using an audio dataset consisting of digits pronounced by six persons. The experimental results reveal that the proposed model with multiple readouts outperforms a conventional model with a single readout.

1. Introduction

Reservoir computing (RC) [1, 2] is a kind of recurrent neural network that has attracted attention in recent years because of its low training cost and potential for hardware implementation via dedicated circuits [3, 4] and physical RC [5, 6]. RC consists of a reservoir part that receives time-series inputs and non-linearly converts them to high-dimensional spaces to represent spatio-temporal patterns of the inputs and a readout part that picks up some of the patterns from the reservoir part to analyze inputs and generates outputs. The main advantage of RC is that its weight connections except in the readout are fixed. As a result, its training requires a smaller amount of data and a lower computational cost compared to deep neural networks. Therefore, RC is suitable for edge AI systems that have limited computational resources and execute training without cloud computing.

The readout of RC is mostly implemented by a linear model (single-layer perceptron) and, therefore, the capability to adapt the training data of the readout is limited. To enhance the training capability of RC, we propose an RC model with multiple readouts that distributes the training of one readout so that each readout can focus on specific kinds of training data. This method can be regarded as a kind of ensemble learning to enhance the RC generalization performance. Simply increasing the number of readouts is inefficient for edge AI systems because it consumes memory resources limited in the systems. This study introduces a self-organizing function that enables the use of the same readout for similar data and different readouts for dissimilar data.

2. Echo state network

To implement the proposed RC with multiple readouts, this study utilizes echo state networks (ESNs) as RC implementations. This section describes the structure and the information processing of ESNs.

An ESN has input, reservoir, and readout layers with N_{in}, N_{res}, and N_{out} nodes, respectively. The reservoir layer contains connections from the input layer and recurrent connections in its layer. The readout layer linearly converts the states of the reservoir layer and generates output signals.

When an input signal $\boldsymbol{u}(t) \in \mathbb{R}^{N_{\text{in}}}$ at time t is given, a state of the reservoir layer $\boldsymbol{x} \in \mathbb{R}^{N_{\text{res}}}$ is updated as follows:

$$\boldsymbol{x}(t) = f(W_{\text{in}}\boldsymbol{u}(t) + W_{\text{res}}\boldsymbol{x}(t-1)), \tag{1}$$

where $W_{\text{in}} \in \mathbb{R}^{N_{\text{res}} \times N_{\text{in}}}$ and $W_{\text{res}} \in \mathbb{R}^{N_{\text{res}} \times N_{\text{res}}}$ are a weight connection between the input and reservoir layers and a recurrent weight connection in the reservoir layer, respectively. f indicates a nonlinear function and a hyperbolic tangent function is often used for this function.

An output of the network $\boldsymbol{y}(t) \in \mathbb{R}^{N_{\text{out}}}$ is generated by the readout layer as follows:

$$\boldsymbol{y}(t) = W_{\text{out}}\boldsymbol{x}(t), \tag{2}$$

where $W_{\text{out}} \in \mathbb{R}^{N_{\text{out}} \times N_{\text{res}}}$ is a weight connection between the reservoir and readout layers. If a target signal $\boldsymbol{z}(t) \in \mathbb{R}^{N_{\text{out}}} (1 \leq t \leq T)$ is given, the weight connection W_{out} can be computed by the ridge regression as follows:

$$W_{\text{out}} = ZX^{\top}(XX^{\top} + \lambda I)^{-1}, \tag{3}$$

$$X = [\boldsymbol{x}(1), \boldsymbol{x}(2), ..., \boldsymbol{x}(T)] \in \mathbb{R}^{N_{\text{res}} \times T} \tag{4}$$

$$Z = [\boldsymbol{z}(1), \boldsymbol{z}(2), ..., \boldsymbol{z}(T)] \in \mathbb{R}^{N_{\text{out}} \times T} \tag{5}$$

where λ is a coefficient of the regularization term of the ridge regression, and $I \in \mathbb{R}^{N_{\text{res}} \times N_{\text{res}}}$ is an identity matrix.

As mentioned above, only the weight connection between the reservoir and readout layers W_{out} is optimized in the training, whereas other weight connections are fixed. Therefore, the capability of the ESN to adapt training data only depends on the readout layer, which is a single linear model, and adapting complex datasets is difficult.

ORCID iDs Yuichiro Tanaka: 0000-0001-6974-070X, Hakaru Tamukoh: 0000-0002-3669-1371

3. Proposed method

This study proposes an RC model with multiple readouts that distribute readout training to enhance the RC training capability. Figure 1 shows the structure of the RC model, which consists of an input layer, a reservoir layer, multiple readouts, and an input similarity map that is implemented by a self-organizing map (SOM) [7]. The SOM consists of nodes aligned on a two-dimensional grid. The number of SOM nodes corresponds to the number of multiple readouts, and each SOM node is associated with each readout.

The input similarity map is used for input data classification to distribute the readout training. For the classification, an unsupervised learning method is desired to avoid additional data labeling for the similarity map, which requires huge man-hours. As one of the unsupervised learning methods, this study adopts the SOM for the input similarity map.

The SOM receives an input signal and classifies it to decide a winner node c as follows:

$$c = \arg\min_i \|\boldsymbol{u}_{\text{concat}} - \boldsymbol{m}_i\| \tag{6}$$

$$\boldsymbol{u}_{\text{concat}} = [\boldsymbol{u}(1)^\top, \boldsymbol{u}(2)^\top, ..., \boldsymbol{u}(T)^\top]^\top \in \mathbb{R}^{N_{\text{in}}T} \tag{7}$$

where i is an index of the SOM node, and $\boldsymbol{m}_i \in \mathbb{R}^{N_{\text{in}}T}$ is a reference vector of i-th SOM node. $\boldsymbol{u}_{\text{concat}}$ is generated by concatenating the input vectors of all time steps, and the winner node of the SOM that has the nearest reference vector to $\boldsymbol{u}_{\text{concat}}$ is selected in this process.

The reference vectors of the SOM are optimized by the unsupervised competitive learning method as follows:

$$\boldsymbol{m}_i^{\text{new}} = \boldsymbol{m}_i + \alpha h_i(\boldsymbol{u}_{\text{concat}} - \boldsymbol{m}_i) \tag{8}$$

$$h_i = \exp(-d_i^2/2\sigma^2) \tag{9}$$

where $\boldsymbol{m}_i^{\text{new}}$ is the updated reference vector generated after this process. α is a learning rate and $h_{i,c}$ is a neighborhood coefficient depending on the distance d_i between the i-th node and the winner node on the SOM grid. σ indicates the width of the neighborhood coefficient affection.

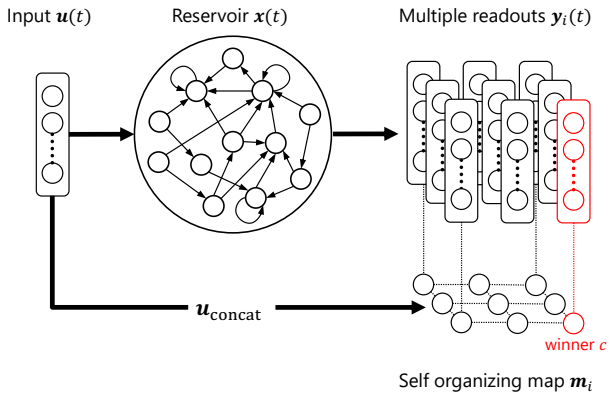

Figure 1: Reservoir computing model with multiple readouts.

The reservoir layer of the model is the same as the reservoir of ESNs, and therefore, the layer receives input signals and updates its state using Eq. (1). Each of the multiple readouts with a weight connection from the reservoir layer $W_{\text{out},i}$ generates an output as follows:

$$\boldsymbol{y}_i(t) = W_{\text{out},i}\boldsymbol{x}(t) \tag{10}$$

where $\boldsymbol{y}_i(t)$ is the i-th readout output at time t. Finally, an output of the readout associated with the winner node (winner readout) $\boldsymbol{y}_c(t)$ is adopted as the network output.

Weight connections of the winner readout and its neighboring readouts are updated in the training phase. The training is designed to strengthen the adaptation of the winning readout to the data that triggers the associated SOM node to win and to provide a rough adaptation of the neighboring readouts to the same data. This training approach enables the winning readout to specialize in data from a specific domain, while the neighboring readouts generalize to data from multiple domains.

To control the readouts' adaptation, this study proposes adjusting the amount of training data based on the neighborhood coefficient of the SOM as shown in Fig. 2. This method prepares memories associated with readouts and stocks the training data in the memories. When T steps input signals and T steps target signals are given to the model, T steps reservoir states $[\boldsymbol{x}(1), \boldsymbol{x}(2), ..., \boldsymbol{x}(T)]$ are obtained. Here, all reservoir states $[\boldsymbol{x}(1), \boldsymbol{x}(2), ..., \boldsymbol{x}(T)]$ and target signals $[\boldsymbol{z}(1), \boldsymbol{z}(2), ..., \boldsymbol{z}(T)]$ are stocked in the memory for the winner readout, whereas $\lfloor h_i T \rfloor$ steps reservoir states $[\boldsymbol{x}(1), ..., \boldsymbol{x}(\lfloor h_i T \rfloor)]$ and the target signals $[\boldsymbol{z}(1), ..., \boldsymbol{z}(\lfloor h_i T \rfloor)]$ are stocked in the memories for the remaining readouts. After feeding all of the training data to the model, each weight connection is computed by the ridge regression shown in Eq. (3) using each stocked data.

4. Experiment

4.1. Data

This study conducted an experiment to evaluate the performance of the proposed RC model using the free spoken digit dataset (FSDD) [8]. This dataset comprises 3,000 audio data of digits ("zero" to "nine") pronounced by six persons and recorded at 8kHz (50 audio data of each digit per person is included). We divided the dataset into training and validation data, with 90% of the data being used for training and 10% for validation.

Before feeding audio data from the FSDD to the model, we used Lyon's auditory model [9] to convert the audio data to a cochleagram, which is a time series of intensities of quantized frequency channels. In this experiment, each cochleagram had 64 frequency channels and 100-time steps so that $N_{\text{in}} = 64$, $T = 100$.

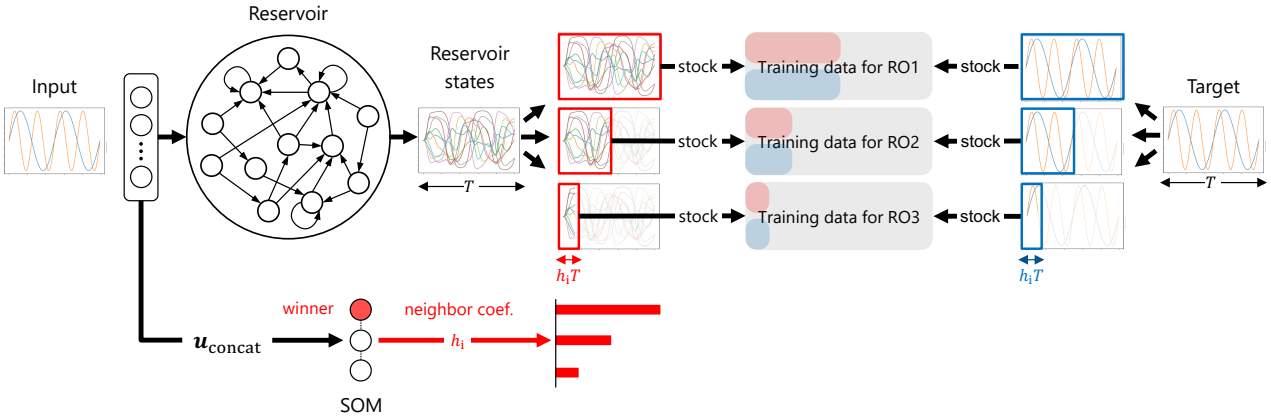

Figure 2: Training of multiple readouts (in the case of the number of readouts is three).

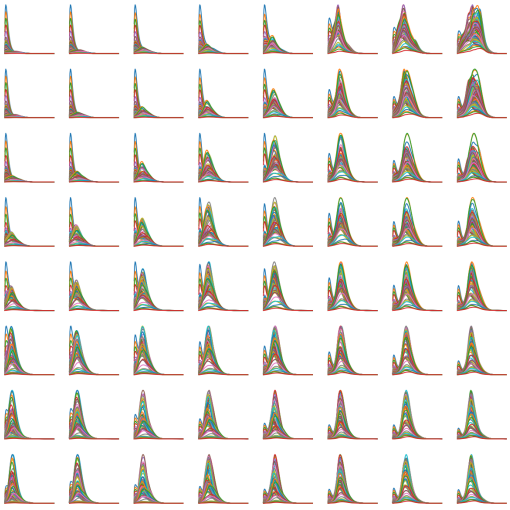

Figure 3: Reference vectors of the SOM after the training.

4.2. Similarity map

This study constructed a SOM whose grid size was 8×8 and fed training data from the FSDD to the SOM. The input signals from the FSDD were concatenated using Eq. (7) so that the concatenated vector dimension $N_{\mathrm{in}}T$ was 6,400. The learning rate of the SOM α and the width of the neighborhood coefficient of the SOM σ varied with time during the SOM training: the learning rate α started from 0.1 and decreased monotonously until 0.001, and the width σ started from 3.0 and decreased monotonously until 0.1.

Figure 3 shows the reference vectors of the SOM after the training, which are aligned on the 8×8 grid. The horizontal axis of each reference vector indicates time steps, the vertical axis indicates intensities, and each line corresponds to the intensity of each frequency channel of the cochleagram.

Table 1: Accuracies of the ESN with a single readout and ESN with multiple readouts in the speaker classification task.

	ESN with single readout	ESN with multiple readout
$N_{\mathrm{res}} = 100$	86.6%	94.7%
$N_{\mathrm{res}} = 250$	93.5%	96.4%
$N_{\mathrm{res}} = 500$	94.5%	97.8%

4.3. Classification tasks

This study conducted two types of sound classification tasks using the FSDD: a speaker classification task and a digit classification task. We fed training data from the FSDD to the reservoir layer and stocked the reservoir states with the target signals based on the neighborhood coefficient h_i in the memories. Here, we used the trained SOM shown in Fig. 3 to determine h_i and set the parameter σ as 1.0. After feeding all of the training data, we computed weight connections of the multiple readouts by the ridge regression setting λ as 0.1. In this experiment, we investigated the test accuracy when the number of the reservoir nodes N_{res} was set to 100, 250, and 500 and compared the accuracy between an ESN with a single readout and an ESN with multiple readouts.

Tables 1 and 2 show the accuracies achieved by the ESN with a single readout and the ESN with multiple readouts in the speaker classification task and the digit classification task, respectively. The ESN with multiple readouts outperformed the ESN with a single readout. We also investigated the performance of a support vector machine (SVM) [10] for both tasks. The SVM achieved an accuracy rate of 95.3% for the speaker classification task and an accuracy rate of 64.7% for the digit classification task.

Table 2: Accuracies of the ESN with a single readout and ESN with multiple readouts in the digit classification task.

	ESN with single readout	ESN with multiple readout
$N_{\text{res}} = 100$	78.7%	83.0%
$N_{\text{res}} = 250$	85.3%	90.1%
$N_{\text{res}} = 500$	89.8%	92.5%

5. Conclusion

This study proposes an RC model with multiple readouts that distribute readout training to enhance the training capability of RC. This study conducted a sound classification task using the FSDD to evaluate the proposed RC model, and the experimental result revealed that the proposed RC model outperformed the conventional ESN.

Because the proposed RC model has multiple readouts, the model is expected to be effective in avoiding catastrophic forgetting [11] in the context of continual learning [12]. If some additional training data is given, an optimized readout of the RC model obtained from the previous training is overwritten and the RC model may not perform the previously trained task well. Conversely, the proposed RC model distributes the training of readouts so that the same readout is used for similar data and different readouts are used for dissimilar data. In this way, the overwriting parameters from the previous training can be avoided.

The proposed method is expected to be utilized not only in sound recognition tasks but also in other tasks such as reinforcement learning [13]. The introduction of the proposed structure in a reservoir-based reinforcement learning model [14] is expected to enhance the performance of the model because catastrophic forgetting can be avoided and transfer learning from knowledge obtained from previously trained episodes may be possible.

Acknowledgments

This paper is based on results obtained from the project JPNP16007 commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

[1] H. 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, 2001.

[2] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural computation*, vol. 14, no. 11, pp. 2531–2560, 2002.

[3] K. Honda and H. Tamukoh, "A hardware-oriented echo state network and its fpga implementation," *Journal of Robotics, Networking and Artificial Life*, vol. 7, pp. 58–62, 2020.

[4] I. Kawashima, Y. Katori, T. Morie, and H. Tamukoh, "An area-efficient multiply-accumulation architecture and implementations for time-domain neural processing," in *2021 International Conference on Field-Programmable Technology (ICFPT)*, 2021.

[5] K. Nakajima, "Physical reservoir computing—an introductory perspective," *Japanese Journal of Applied Physics*, vol. 59, no. 6, p. 060501, may 2020.

[6] Y. Usami, B. van de Ven, D. G. Mathew, T. Chen, T. Kotooka, Y. Kawashima, Y. Tanaka, Y. Otsuka, H. Ohoyama, H. Tamukoh, H. Tanaka, W. G. van der Wiel, and T. Matsumoto, "In-materio reservoir computing in a sulfonated polyaniline network," *Advanced Materials*, vol. 33, no. 48, p. 2102688, 2021.

[7] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[8] Z. Jackson, C. Souza, J. Flaks, Y. Pan, H. Nicolas, and A. Thite, "Jakobovski/free-spoken-digit-dataset," 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1342401

[9] R. Lyon, "A computational model of filtering, detection, and compression in the cochlea," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 7, 1982, pp. 1282–1285.

[10] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[11] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," ser. Psychology of Learning and Motivation, G. H. Bower, Ed. Academic Press, 1989, vol. 24, pp. 109–165.

[12] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[13] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. A Bradford Book, 2018.

[14] M. Inada, Y. Tanaka, H. Tamukoh, K. Tateno, T. Morie, and Y. Katori, "A reservoir based q-learning model for autonomous mobile robots," in *2020 International Symposium on Nonlinear Theory and Its Applications (NOLTA2020)*, 2020, pp. 213–216.