

LETTER

Reservoir-Based 1D Convolution: Low-Training-Cost AIYuichiro TANAKA^{†a)} and Hakaru TAMUKOH^{†,††}, *Members*

SUMMARY In this study, we introduce a reservoir-based one-dimensional (1D) convolutional neural network that processes time-series data at a low computational cost, and investigate its performance and training time. Experimental results show that the proposed network consumes lower training computational costs and that it outperforms the conventional reservoir computing in a sound-classification task.

key words: edge computing, reservoir computing, reservoir-based convolution, voice recognition

1. Introduction

Most current artificial intelligence (AI) technologies are based on deep learning (DL) [1], [2]. DL achieves state-of-the-art results in various tasks because it uses a significant amount of training data to maximize performance, and high-performance computers with graphics processing units (GPUs), to accelerate the computation. However, it is difficult to apply DL in edge AI because the amount of training data is insufficient. Additionally, because the power of the edge systems is limited, GPUs, which consume considerable power, are unsuitable. Thus, a low-training-cost AI that requires little training data and has low computational costs, resulting in a low-power implementation, is necessary for edge AI.

Reservoir computing (RC) [3], [4] can be a solution for low-training-cost AI because only part of the weight connections has plasticity, whereas normal neural networks optimized by backpropagation [5] update all weight connections during training. As a reservoir-based approach, Tanaka and Tamukoh proposed a reservoir-based convolutional neural network (CNN) [6], which is an extended version of RC and an effective neural network to process images. The reservoir-based CNN outperforms existing reservoir-based approaches in some image-recognition tasks while maintaining a low training cost.

Toward the realization of a low-training-cost AI for not only image processing but also time-series data processing, this study proposes a reservoir-based one-dimensional (1D)

convolution, an extended version of the previously proposed reservoir-based two-dimensional (2D) convolution, and investigates its classification performances for time-series data and its computational costs during training.

2. Proposed Method

The reservoir-based 1D convolution operation uses several reservoirs, as shown in Fig. 1. When an input with N_{ch} channels is provided to the reservoir-based 1D convolution layer, the reservoirs receive a region of interest (ROI) of width T from the input, and process the ROI as time-series data $\mathbf{u}(t) \in \mathbb{R}^{N_{\text{ch}}}$, where t indicates a discrete time-step ($t = 1, 2, \dots, T$), according to the following equation,

$$\mathbf{x}(t) = (1 - \delta)\mathbf{x}(t - 1) + \delta f(W_{\text{ch}}\mathbf{u}(t) + W_{\text{res}}\mathbf{x}(t - 1)) \quad (1)$$

where $\mathbf{x}(t) \in \mathbb{R}^{N_{\text{res}}}$ is a state of one of the reservoirs; its initial state is set as $\mathbf{x}(0) = \mathbf{0}$. $W_{\text{ch}} \in \mathbb{R}^{N_{\text{res}} \times N_{\text{ch}}}$ and $W_{\text{res}} \in \mathbb{R}^{N_{\text{res}} \times N_{\text{res}}}$ denote a weight connection between the ROI and reservoir and a recurrent weight connection in the reservoir, respectively. f indicates a nonlinear function; a hyperbolic tangent function was used in this study. δ ($0 < \delta < 1$) indicates a leak rate, which controls the updating speed of the reservoir. After feeding the ROI to the reservoirs, only the final states of the reservoirs are adopted as elements of feature maps. If the number of reservoirs in the operation is R , the feature maps have $R \times N_{\text{res}}$ channels. The ROI is shifted, and the above computation is repeated in the same manner as the 1D convolutional operation to compute the entire feature map. Therefore, the reservoir-based 1D convolution can be seen as a time-domain operating version of the ordinary convolution.

Reservoirs in the operation have several leak rates to extract various features from inputs. As mentioned in [6],

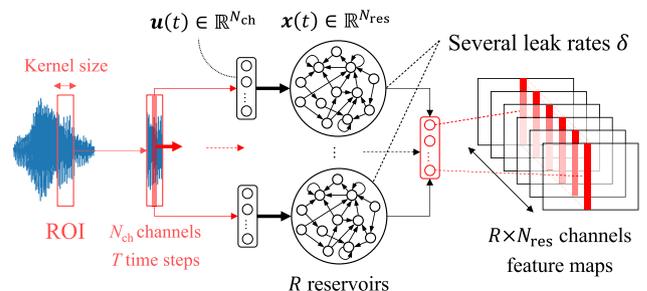


Fig. 1 Reservoir-based 1D convolution.

Manuscript received June 14, 2023.

Manuscript revised August 28, 2023.

Manuscript publicized September 11, 2023.

[†]The authors are with Research Center for Neuromorphic AI Hardware, Kyushu Institute of Technology, Kitakyushu-shi, 808-0196 Japan.

^{††}The author is with Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology, Kitakyushu-shi, 808-0196 Japan.

a) E-mail: tanaka-yuichiro@brain.kyutech.ac.jp

DOI: 10.1587/transfun.2023EAL2050

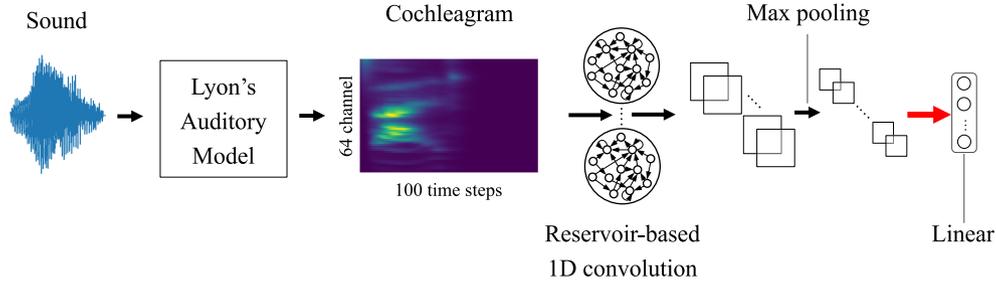


Fig. 2 Network construction.

a reservoir with a small leak rate works as a rough feature extractor because the reservoir updating speed is slow and the reservoir cannot follow the fast changes in the inputs. Conversely, a reservoir with a large leak rate works as a fine feature extractor. Because of such a structure, the reservoir-based 1D convolution has a stronger feature-extracting function compared to the ordinary RC.

3. Experiment

We constructed a reservoir-based 1D CNN consisting of a reservoir-based 1D convolution, max-pooling, and linear layers, as shown in Fig. 2, where only the weight connections between the max-pooling and linear layers (indicated by the red arrow in Fig. 2) have plasticity, and verified the network performance in sound-classification tasks. Table 1 shows the parameters used in the network where kernel size and stride size indicate the ROI width T and the amount of shift of ROI, respectively. The reservoir-based 1D convolution layer had five reservoirs, each with 30 nodes. The leak rate of the i -th reservoir was set as $0.2 \times (i - 1) + 0.1$.

We used the Free Spoken Digit Dataset (FSDD) [7] as a dataset for verification. This dataset consists of 3,000 audio data of English digits spoken by six persons, recorded at 8 kHz. We used 300 FSDD data samples as test data and the remaining 2,700 data samples as training data. The audio data were preprocessed using Lyon's auditory model [8] and converted to a cochleagram, a time-series data of signal intensities of quantized frequency channels. In our experiment, each cochleagram had 64 channels and 100 time-steps (the preprocessing included decimation).

We trained the network using the following procedure: We fed the training data into the network and checked the output from the max-pooling layer. By using the output of the max-pooling layer and target signals where labels were represented by one-hot vectors, we computed an optimized weight connection between the max-pooling and linear layers W_{lin} using ridge regression, as follows,

$$W_{\text{lin}} = ZM^T(MM^T + \lambda I)^{-1} \quad (2)$$

$$M = [\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_j, \dots, \mathbf{m}_{2700}] \quad (3)$$

$$Z = [\mathbf{z}_1, \mathbf{z}_1, \dots, \mathbf{z}_j, \dots, \mathbf{z}_{2700}] \quad (4)$$

where \mathbf{m}_j and \mathbf{z}_j are the j -th output vector from the max-pooling layer and j -th target signal, respectively. λ is a

Table 1 Parameters of the reservoir-based 1D CNN.

Layer	Parameters
Reservoir-based 1D convolution	Kernel size: 10 Stride size: 5
Max-pooling	Kernel size: 4 Stride size: 4
Linear	Input size: 600 Output size: 10 or 6

Table 2 Comparison of classifiers for FSDD (digit classification) in terms of accuracy and training time.

	Accuracy [%]	Training time [s]
Support vector machine	69.3	25.3
Random forest (50 trees)	95.3	7.67
Random forest (100 trees)	96.1	15.2
Echo state network	91.5	16.3
Reservoir-based 1D CNN	95.5	5.21

Table 3 Comparison of classifiers for FSDD (speaker classification) in terms of accuracy and training time.

	Accuracy [%]	Training time [s]
Support vector machine	95.3	11.0
Random forest (50 trees)	99.1	6.56
Random forest (100 trees)	98.9	13.3
Echo state network	98.5	18.0
Reservoir-based 1D CNN	98.2	5.40

coefficient of the regularization term of the ridge regression and I is an identity matrix. Additionally, we measured the computation time during training when Intel Xeon @ 2.20 GHz was used for processing. Because FSDD has two types of labels — digit and speaker — we conducted a digit-classification task (ten-class classification) and a speaker-classification task (six-class classification) in this experiment.

We verified the accuracy of the trained network using the test data. For comparison, we also verified the accuracy and training time of a support vector machine (SVM) [9], random forest [10] with 50 and 100 trees, and an echo state network (ESN) [3] with 600 reservoir nodes, using the same conditions as those of the reservoir-based 1D CNN. The SVM and random forest were implemented using scikit-learn [11]. Tables 2 and 3 show the results of the digit and speaker-classification tasks.

To verify the feature-extracting function of the reservoir-based 1D convolution, we visualized the feature

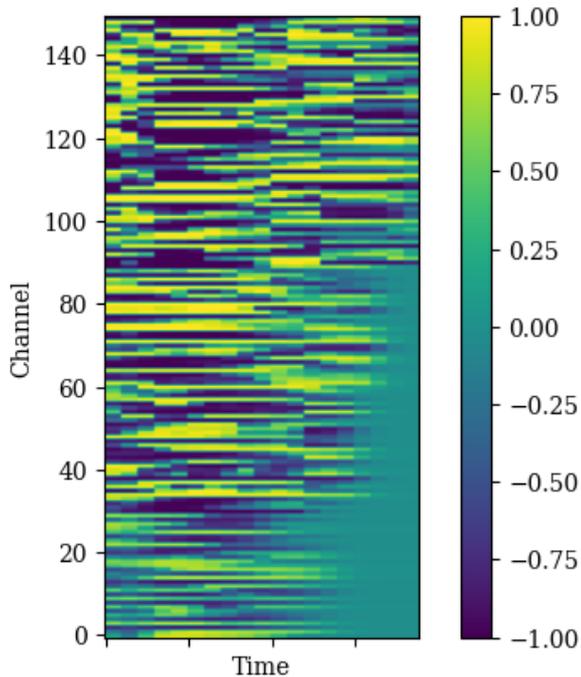


Fig. 3 Feature map generated by the reservoir-based 1D convolution layer receiving a cochleagram of FSDD.

map generated by the reservoir-based 1D convolution layer receiving a cochleagram of FSDD as shown in Fig. 3. On the horizontal axis of the figure is the time of the input and the channels are on the vertical axis. Channels 0 to 29 correspond to the output of the reservoir with a leak rate of 0.1, Channels 30 to 59 correspond to the output of the reservoir with a leak rate of 0.3, \dots , and Channels 120 to 149 correspond to the output of the reservoir with a leak rate of 0.9. As shown in the figure, reservoirs with low leak rates tended to extract rough features, and reservoirs with high leak rates tended to extract fine features from the input.

4. Discussion

In the case of the digit-classification task, the accuracy of the reservoir-based 1D CNN was better than that of the SVM, random forest with 50 trees, and ESN, but not as good as that of the random forest with 100 trees, as shown in Table 2. In the case of the speaker-classification task, the accuracy of the reservoir-based 1D CNN was better than that of the SVM, but less than that of the random forests and ESN, as shown in Table 3. In both tasks, the reservoir-based 1D CNN had the shortest training time among the verified classifiers.

Although ridge regression was used to optimize both the reservoir-based 1D CNN and ESN, and the number of optimized parameters was common in both networks, the training time of the reservoir-based 1D CNN was shorter than that of the ESN because of the difference in the matrix sizes used in the optimizations. The ESN used all reservoir states (each reservoir state was a 600-dimensional vector) when the input time-series data was being fed (2,700 data samples, each of the 100 time-steps) for the optimization. Therefore,

the size of the matrix M used in the ridge regression was $600 \times 270,000$. Conversely, the reservoir-based 1D CNN used only the final states of the reservoirs so that the size of the matrix M was $600 \times 2,700$.

A possible reason for the accuracy of the reservoir-based 1D CNN being less than that of the random forests and ESN in the speaker-classification task is overfitting. We checked the weight connections between the max-pooling and linear layers W_{lin} after training for both cases of the digit- and speaker-classification tasks, and found that the maximum value of W_{lin} for the speaker-classification task was 1.3 times as large as that for the digit-classification task. Because overfitting increases the values of the weight connections in general, we concluded that the reservoir-based 1D CNN overfitted the training data in the speaker-classification task.

5. Conclusion

This study proposed a reservoir-based 1D convolutional operation and a neural network using the operation. The experimental results showed that the computation time for training the proposed network was shorter than that for the conventional RC. Additionally, the network had a strong feature-extracting function and achieved accuracies of 95.5% and 98.2% in the digit-classification and speaker-classification tasks, respectively.

Because the proposed network consumes low computational costs in training, the network could be applied to edge AI where not only inference but also training must be executed on the edge. Although this study verified the network performance in sound-classification tasks, the network is expected to be used in other tasks where time-series data are provided and high performance is expected.

Hardware implementation is an effective solution to realize a low-power system. Several studies have proposed dedicated hardware designs for ESNs using field-programmable gate arrays [12]–[15]. Therefore, the reservoir-based 1D CNN can be implemented on field-programmable gate arrays using these designs. Moreover, physical reservoir implementation [16]–[20] has been studied to achieve further reduction in power, compared with that achieved using the conventional semiconductor-based approaches. For example, Tanaka et al. proposed the hardware implementation of reservoir-based 2D convolution using a nanomaterial-based device [21], [22]; this implies that nanomaterial-based physical reservoir implementation of the reservoir-based 1D CNN is possible.

Acknowledgments

This paper is based on the results obtained from a project, JPNP16007, commissioned by the New Energy and Industrial Technology Development Organization (NEDO) and supported by JSPS KAKENHI Grant Number 21K21318, 22K17968, 23H03468.

References

- [1] G.E. Hinton, S. Osindero, and Y.W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol.18, no.7, pp.1527–1554, 2006.
 - [2] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol.86, no.11, pp.2278–2324, 1998.
 - [3] 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.
 - [4] 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.
 - [5] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning representations by back-propagating errors," *Nature*, vol.323, pp.533–536, 1986.
 - [6] Y. Tanaka and H. Tamukoh, "Reservoir-based convolution," *Nonlinear Theory and Its Applications, IEICE*, vol.13, no.2, pp.397–402, 2022.
 - [7] Z. Jackson, C. Souza, J. Flaks, Y. Pan, H. Nicolas, and A. Thite, "Jakobovski/free-spoken-digit-dataset," 2018.
 - [8] R. Lyon, "A computational model of filtering, detection, and compression in the cochlea," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp.1282–1285, 1982.
 - [9] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol.20, no.3, pp.273–297, 1995.
 - [10] T.K. Ho, "Random decision forests," *Proc. 3rd International Conference on Document Analysis and Recognition*, pp.278–282, IEEE, 1995.
 - [11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol.12, no. Oct, pp.2825–2830, 2011.
 - [12] M.L. Alomar, V. Canals, N. Perez-Mora, V. Martínez-Moll, and J.L. Rosselló, "FPGA-based stochastic echo state networks for time-series forecasting," *Computational Intelligence and Neuroscience*, vol.2016, 2016.
 - [13] 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.
 - [14] N.S. Huang, J.M. Braun, J.C. Larsen, and P. Manoonpong, "A scalable echo state networks hardware generator for embedded systems using high-level synthesis," *2019 8th Mediterranean Conference on Embedded Computing (MECO)*, pp.1–6, IEEE, 2019.
 - [15] M.L. Alomar, E.S. Skibinsky-Gitlin, C.F. Frasser, V. Canals, E. Isern, M. Roca, and J.L. Rosselló, "Efficient parallel implementation of reservoir computing systems," *Neural Comput. & Applic.*, vol.32, no.7, pp.2299–2313, 2020.
 - [16] G. Tanaka, T. Yamane, J.B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, "Recent advances in physical reservoir computing: A review," *Neural Networks*, vol.115, pp.100–123, 2019.
 - [17] K. Nakajima, "Physical reservoir computing — An introductory perspective," *Jpn. J. Appl. Phys.*, vol.59, no.6, p.060501, May 2020.
 - [18] G. Van der Sande, D. Brunner, and M.C. Soriano, "Advances in photonic reservoir computing," *Nanophotonics*, vol.6, no.3, pp.561–576, 2017.
 - [19] J. Torrejon, M. Riou, F.A. Araujo, S. Tsunegi, G. Khalsa, D. Querlioz, P. Bortolotti, V. Cros, K. Yakushiji, A. Fukushima, H. Kubota, S. Yuasa, M.D. Stiles, and J. Grollier, "Neuromorphic computing with nanoscale spintronic oscillators," *Nature*, vol.547, no.7664, pp.428–431, 2017.
 - [20] K. Nakajima, H. Hauser, T. Li, and R. Pfeifer, "Information processing via physical soft body," *Scientific Reports*, vol.5, no.1, pp.1–11, 2015.
 - [21] 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-material reservoir computing in a sulfonated polyaniline network," *Advanced Materials*, vol.33, no.48, p.2102688, 2021.
 - [22] Y. Tanaka, Y. Usami, H. Tanaka, and H. Tamukoh, "In-material reservoir implementation of reservoir-based convolution," *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp.1–5, 2023.
-