IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 4, 2021, accepted December 1, 2021, date of publication December 10, 2021,
date of current version December 23, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3134700

Deep Residual Learning With Dilated Causal Convolution Extreme Learning Machine

AKIRA SASOU, (Member, IEEE)
National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba 305-8560, Japan
e-mail: a-sasou@aist.go.jp

This work was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant JP20H04218, and in part by the New Energy and Industrial Technology Development Organization (NEDO) under Project JPNP20006.

ABSTRACT A feedforward neural network with random weights (RW-FFNN) uses a randomized feature map layer. This randomization enables the optimization problem to be replaced by a standard linear least-squares problem, which offers a major advantage in terms of training speed. An extreme learning machine (ELM) is a well-known RW-FFNN that can be implemented as a single-hidden-layer feedforward neural network. However, for a large dataset, owing to the shallow architecture, such an ELM typically requires a very large number of nodes in a single hidden layer to achieve a sufficient level of accuracy. In this paper, we propose a deep residual learning method with a dilated causal convolution ELM (DRLDCC-ELM). The baseline layer performs feature mapping to predict the target features based on the input features. The subsequent residual-compensation layers then iteratively remodel the uncaptured prediction errors in the previous layer. The proposed network architecture also adopts dilated causal convolution based on the ELM in each layer to effectively expand the receptive field of the multilayer network. The results of experiments involving acoustic scene classification of daily activities in a home environment confirmed that the proposed DRLDCC-ELM outperforms the previously proposed residual-compensation ELM and deep-residual-compensation ELM methods. We also confirmed that the generalization capability of the proposed DRLDCC-ELM tends to be superior to that of convolutional neural network-based models, especially for a large number of parameters.

INDEX TERMS Feedforward neural network with random weights, extreme learning machine, deep residual learning, dilated causal convolution, acoustic scene classification.

I. INTRODUCTION

A feedforward neural network with random weights (RW-FFNN) uses a randomized feature map layer that is defined independently of the training data. This randomization offers a major advantage in that optimization becomes a standard linear least-squares problem, which can be solved using a single analytical step implemented efficiently in most linear algebra libraries. Ideas similar to the RW-FFNN have been proposed many times, in different forms, over the last several decades [1]. The current form of the RW-FFNN was originally introduced as a random vector functional link network [2]. Recently, an RW-FFNN in the form of a single-hidden-layer feedforward neural network (SLFN) has been proposed and is known as an extreme learning machine (ELM) [3]. In an ELM, the weights between the input layer and the hidden layer, and the biases of the hidden layer are randomly generated and do not need to be adjusted. The weights between the hidden and output layers were analytically determined using the least-squares method. Over the last decade, researchers have devoted attention to comparing ELMs with other machine learning algorithms, including support vector machine [50], random neural network [51]–[54], radial basis function neural network [55]–[58], and other deep learning methods. Generally speaking, ELMs are faster than those methods and their variants tend to be more robust [4]. ELMs have been applied to many learning tasks for classification, clustering, and regression. Because of their superiority in terms of training speed, accuracy, and generalization, ELMs have been used in fields such as medicine [14]–[29], chemistry [30]–[32], economics [33]–[40], transportation [41]–[45], robotics [46]–[49], and so on [4].

The associate editor coordinating the review of this manuscript and approving it for publication was Mostafa Rahimi Azghadi.

However, for a large dataset, owing to the shallow architecture, an ELM typically requires a very large number of nodes in a single hidden layer to achieve a sufficient level of accuracy, which makes the process computationally intensive. To overcome this, multilayer ELMs have been proposed in recent years. Kasun *et al.* proposed an ELM autoencoder (ELM-AE) [5] in which a multilayer ELM was constructed by iteratively stacking several layers. This ELM-AE was proposed mainly for classification problems rather than regression problems. Zhang *et al.* proposed a residual-compensation ELM (RC-ELM) for regression problems, which compensated for prediction errors [6]. The RC-ELM employs a multilayer structure consisting of a baseline layer and several residual-compensation layers. The baseline layer constructs a feature map that predicts the target features from the input features. The residual-compensation layers iteratively remodel the uncaptured prediction errors in the previous layer, where the same input features fed into the baseline layer are repeatedly fed into the residual-compensation layers. Additionally, the output of each individual layer is not fed into the input of another layer. Therefore, an RC-ELM has an ensemble network structure rather than a deep network structure. However, it has been reported [8] that an RC-ELM cannot perform effective error correction and tends to cause overfitting after the stacking of three residual-compensation layers. Chen *et al.* proposed a deep-residual-compensation ELM (DRC-ELM) to improve the robustness and generalization capabilities of an RC-ELM [8]. A DRC-ELM borrows the basic idea from a residual neural network [9] and has a deep network structure. The architecture of a DRC-ELM is slightly different from that of a residual neural network; in the former, the predicted target features for the previous layer are concatenated with the original features fed into the baseline layer, and the new concatenated features are then fed into the following layer. The input features fed into the baseline layer are thus repeatedly fed into the residual-compensation layers, as in the case of an RC-ELM. Chen *et al.* applied a DRC-ELM to predict gold prices based on those for the five previous trading days and several other indices, and demonstrated the effectiveness of the method [8]. However, in the case of time-series predictions based on past observations over a larger area, DRC-ELM is not practical because of the intensive calculations required.

In this paper, we propose a modified architecture for deep residual learning using a dilated causal convolution ELM that is applicable to learning tasks such as recognition, classification, and regression of a time series of feature vectors. The proposed network architecture effectively utilizes past input features over a larger area and offers improved robustness and generalization capabilities. Our architecture adopts a dilated causal convolution based on an ELM in each layer to effectively expand the receptive field of the multilayer network. This idea stems from WaveNet [11]. In a dilated convolution, a filter is applied to an area larger than the filter width by skipping the input features with a certain step size. The dilation step size can be changed layer by layer. Stacked dilated convolutions enable networks to have very large receptive fields using only a few layers, while preserving the input resolution throughout the network as well as the computational efficiency [11]. In the baseline layer, the original input and time-delayed features are concatenated and fed into the ELM to achieve dilated causal convolution, where the delay time corresponds to the dilation size of the ELM filter with randomized weights, and is then trained to directly predict the original target features. In the residual-compensation layers, the original input features and the prediction of the previous stacked layer are concatenated, and the results are then concatenated with the time-delayed features and fed into the ELM. By using a combination of both the original input features and the predicted target features as the ELM input, together with the corresponding prediction residuals, the ELM for the residual-compensation layer can effectively learn the prediction characteristics of the previously stacked layers. This is expected to allow the ELM to accurately infer the residual of the previous layer. In addition, because the proposed method executes the dilated causal convolutions with different dilation sizes for each layer, this is expected to improve the learning efficiency of the ELM compared with a DRC-ELM that repeatedly feeds the same original input features into each layer. Hereinafter, we refer to the proposed model as a deep residual learning with dilated causal convolution ELM (DRLDCC-ELM).

II. EXTREME LEARNING MACHINE

The ELM in the present study is an RW-FFNN in the form of an SLFN, as depicted in Fig. 1. Here, we assume that the input layer, hidden layer, and output layer consist of M, L, and K nodes, respectively. Let $w_{l,m}$ denote the weight between the m-th node in the input layer and the l-th node in the hidden layer, and let b_l denote the bias at the l-th node in the hidden layer. The output of the l-th node in the hidden layer is then given by

$$g_l\left(\mathbf{x}_t\right) = g\left(\sum_{m=1}^{M} w_{l,m} x_{t,m} + b_l\right), \tag{1}$$

where $\mathbf{x}_t = \left[x_{t,1} \cdots x_{t,M}\right]^T, t = 1, \cdots, N$ represents the input feature vectors, N denotes the number of input feature vectors, and $g_l\left(\cdot\right)$ represents a nonlinear activation function.

The weights and biases were generated randomly. Huang *et al.* [10] proposed a method to orthogonalize the weights so that the network can extract a more complete set of features in comparison with non-orthogonalized weights, and reported that the use of orthogonalized weights can achieve better performance than even their well-trained counterparts. Inspired by this, the proposed method adopts orthonormalized weights obtained by applying the Gram–Schmidt orthonormalization method to randomly generated weights. If the number of nodes in the hidden layer, L, is larger than the number of nodes in the input layer, M, the weight vectors $\mathbf{w}_l = \left[w_{l,1} \cdots w_{l,M}\right]^{\mathrm{T}}, l = 1, \cdots, L$, are grouped into M separate vectors to form the matrix

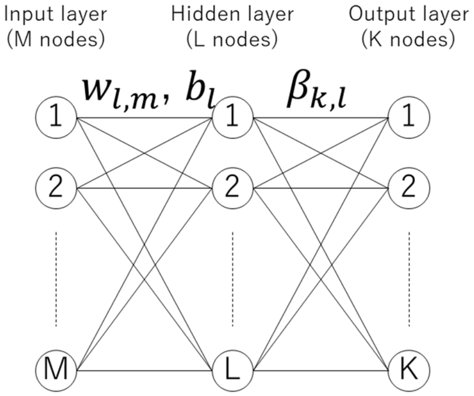


FIGURE 1. Single hidden layer feed forward neural network (SLFN).

$\mathbf{W}_q = \left[\mathbf{w}_{(q-1)M+1} \cdots \mathbf{w}_{qM}\right], q = 1, \cdots, \lfloor L/M \rfloor$. The weight vectors in each matrix are orthonormalized such that $\mathbf{W}_q^{\mathrm{T}}\mathbf{W}_q = \mathbf{I}$. If the remainder of the division L/M exists, $\mathbf{W}_{\lfloor L/M \rfloor+1} = \left[\mathbf{w}_{\lfloor L/M \rfloor M+1} \cdots \mathbf{w}_L\right]$ is also orthonormalized in the limited rank of the matrix. The bias vector $\mathbf{b} = \left[b_1 \cdots b_L\right]^{\mathrm{T}}$ is normalized as $\mathbf{b}^{\mathrm{T}}\mathbf{b} = 1$.

Next, let $\beta_{k,l}$ denote the weight between the l-th node in the hidden layer and the k-th node in the output layer. The predicted output value for the k-th node in the output layer is given by

$$f_k(\mathbf{x}_t) = \sum_{l=1}^{L} \beta_{k,l}\, g_l(\mathbf{x}_t). \tag{2}$$

We define the predicted target feature matrix based on all the predicted target feature vectors $\hat{y}_t = \left[f_1(\mathbf{x}_t) \cdots f_K(\mathbf{x}_t)\right]^{\mathrm{T}}, t = 1, \cdots, N$ as

$$\begin{aligned}\hat{\mathbf{Y}} &= \begin{bmatrix} \hat{\mathbf{y}}_1^{\mathrm{T}} \\ \vdots \\ \hat{\mathbf{y}}_N^{\mathrm{T}} \end{bmatrix} \\ &= \begin{bmatrix} g_1(\mathbf{x}_1) & \cdots & g_L(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ g_1(\mathbf{x}_N) & \cdots & g_L(\mathbf{x}_N) \end{bmatrix} \begin{bmatrix} \beta_{1,1} & \cdots & \beta_{K,1} \\ \vdots & \ddots & \vdots \\ \beta_{1,L} & \cdots & \beta_{K,L} \end{bmatrix} \\ &= \mathbf{H}\boldsymbol{\beta}. \end{aligned} \tag{3}$$

Given a training dataset $(\mathbf{x}_t, \mathbf{y}_t)$, where $\mathbf{y}_t$ denotes an original target feature vector $\mathbf{y}_t = \left[y_{t,1} \cdots y_{t,K}\right]^{\mathrm{T}}$, the optimal weight matrix, $\boldsymbol{\beta}^*$, can be obtained as a solution to a standard regularized least-squares problem [10]:

$$\boldsymbol{\beta}^* = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + C\frac{1}{2}\|\mathbf{H}\boldsymbol{\beta} - \mathbf{Y}\|_2^2, \tag{4}$$

where $\mathbf{Y}$ denotes the original target feature matrix $\mathbf{Y}^{\mathrm{T}} = \left[\mathbf{y}_1 \cdots \mathbf{y}_N\right]$ and C denotes a user-defined scalar for weighting the prediction error term. The closed-form solution of Eq. (4) is given by

$$\boldsymbol{\beta}^* = \begin{cases} \mathbf{H}^{\mathbf{T}}\left(\dfrac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^{\mathbf{T}}\right)^{-\mathbf{1}}\mathbf{Y}, & \text{if } N \le L \\ \left(\dfrac{\mathbf{I}}{C} + \mathbf{H}^{\mathbf{T}}\mathbf{H}\right)^{-\mathbf{1}}\mathbf{H}^{\mathbf{T}}\mathbf{Y}, & \text{if } N > L \end{cases} \tag{5}$$

III. PROPOSED METHOD

The proposed model (DRLDCC-ELM) adopts a dilated causal convolution ELM in each layer to effectively expand the receptive field of a deep-residual-compensation ELM. Figs. 2(a) and 2(b) depict the baseline layer and residual-compensation layer for the DRLDCC-ELM, respectively. In contrast to a residual neural network in which the entire network is simultaneously trained using backpropagation, the DRLDCC-ELM trains each ELM layer by layer from the baseline layer and stacks the trained ELMs in order. Let $\mathbf{i}_t \in \mathbb{R}^p$ and $\mathbf{o}_t \in \mathbb{R}^q$ denote an original input feature vector and an original target feature vector, respectively. During training, the pairs of input and output feature vectors $(\mathbf{i}_t, \mathbf{o}_t), t = 1, \cdots, N$ are given. The following describes the training method for the baseline layer and residual-compensation layers.

In the baseline layer, the original input feature vectors are directly fed into the feature preprocessing block, which applies a time delay and performs concatenation. Let n_1 denote the time-delay parameter that determines the dilation step. The original input feature vector, $\mathbf{i}_t \in \mathbb{R}^p$, and its time-delayed vector, $\mathbf{i}_{t-n_1}$, are concatenated as

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{i}_t \\ \mathbf{i}_{t-n_1} \end{bmatrix} \in \mathbb{R}^{2p}, \tag{6}$$

which is an input feature vector fed into the next ELM. The time-delayed vectors in the range of $t - n_1 < 1$ were set to $\mathbf{i}_{t-n_1} = \mathbf{0}$. The dimension of the ELM input feature vector is $2p$ and each element value of the vector is assigned to each node in the input layer, so the number of nodes in the input layer is also set to $M = 2p$. The ELM in the baseline layer is trained to directly predict the original target feature vector. Thus, the target feature vector of the ELM is given by

$$\mathbf{y}_t = \mathbf{o}_t \in \mathbb{R}^q. \tag{7}$$

Because the dimension of the target feature vector is q, the number of nodes in the output layer is set to $K = q$. The number of nodes in the hidden layer is a user-defined parameter. The training of the ELM is conducted using the training dataset $(\mathbf{x}_t, \mathbf{y}_t)$ given by Eqs. (6) and (7), respectively: We first prepare the randomly generated weight vectors, $\mathbf{w}_l = \left[w_{l,1} \cdots w_{l,M}\right]^{\mathrm{T}}, l = 1, \cdots, L$, and the bias vector, $\mathbf{b} = \left[b_1 \cdots b_L\right]^{\mathrm{T}}$, and then orthonormalize these vectors according to the method described in section II. The output values in the hidden layer are then given by Eq. (1). Matrix $\mathbf{H}$ defined in Eq. (3) is constructed using the output values in the hidden layer. Using this matrix and the original target feature matrix, $\mathbf{Y}$, the optimal weight matrix $\boldsymbol{\beta}^*$, is given by Eq. (5). The ELM's predicted target feature matrix $\hat{\mathbf{Y}}$, can be calculated using Eq. (3) by replacing the weight matrix $\boldsymbol{\beta}$, with the obtained optimal matrix $\boldsymbol{\beta}^*$. Because the ELM in the baseline layer is intended to predict the original target feature vectors, $\mathbf{o}_t$, the target feature vectors, $\hat{\mathbf{y}}_t$, predicted by the ELM are regarded as the original target feature vector prediction, $\hat{\mathbf{o}}_t^1$:

$$\hat{\mathbf{o}}_t^1 = \hat{\mathbf{y}}_t \in \mathbb{R}^q \tag{8}$$

where the superscript 1 indicates that the output feature vector is obtained from the baseline layer.

In the residual-compensation layers, several layers were trained and stacked on the baseline layer. To distinguish the layers, we introduce a layer index denoted by #s, where $s = 1$ indicates the baseline layer and $s > 1$ indicates the residual-compensation layers. In the following, residual compensation layer #s is assumed to be trained. Let $\hat{\mathbf{o}}_t^{s-1} \in \mathbb{R}^q$ denote the predicted original target feature vector for the previous layer, which is to be concatenated with the original input feature vector, $\mathbf{i}_t \in \mathbb{R}^p$, to form a new feature vector, $\mathbf{v}_t^s$, as follows:

$$\mathbf{v}_t^s = \begin{bmatrix} \mathbf{i}_t \\ \hat{\mathbf{o}}_t^{s-1} \end{bmatrix} \in \mathbb{R}^{(p+q)}. \tag{9}$$

This feature vector, $\mathbf{v}_t^s$, is then fed into the feature preprocessing block, which introduces a time delay of n_s and performs concatenation. The ELM input feature vector in this layer is given by

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{v}_t^s \\ \mathbf{v}_{t-n_s}^s \end{bmatrix} \in \mathbb{R}^{2(p+q)}. \tag{10}$$

The dimension of the ELM input feature vector is $2\,(p+q)$, so the number of nodes in the input layer is also $M = 2\,(p+q)$. The ELM in this residual compensation layer is trained to predict the residual feature vector for the previous layer, $\mathbf{r}_t^{s-1}$. Thus, the target feature vector for the ELM is given by

$$\mathbf{y}_t = \mathbf{r}_t^{s-1} = \mathbf{o}_t - \hat{\mathbf{o}}_t^{s-1} \in \mathbb{R}^q. \tag{11}$$

Because the dimension of the target feature vector is q, the number of nodes in the output layer is set to $K = q$. The number of nodes in the hidden layer is a user-defined parameter. The training of the ELM is conducted using the training dataset $(\mathbf{x}_t, \mathbf{y}_t)$ given by Eqs. (10) and (11), respectively: The ELM was trained using the same procedure as that for the baseline layer. Because the ELM in the residual compensation layer is intended to predict the residual feature vectors for the previous layer, $\mathbf{r}_t^{s-1}$, the predicted target feature vectors for the ELM, $\hat{\mathbf{y}}_t$, are regarded as the prediction for the residual feature vector $\hat{\mathbf{r}}_t^{s-1}$:

$$\hat{\mathbf{r}}_t^{s-1} = \hat{\mathbf{y}}_t \in \mathbb{R}^q. \tag{12}$$

Therefore, the prediction for the original target feature vector at this layer, $\hat{\mathbf{o}}_t^s$, is given by

$$\hat{\mathbf{o}}_t^s = \hat{\mathbf{o}}_t^{s-1} + \hat{\mathbf{r}}_t^{s-1} \in \mathbb{R}^q. \tag{13}$$

For S stacked layers, the prediction for the final layer, $\hat{\mathbf{o}}_t^S$, for the original target feature vector can be rewritten by recursively applying Eq. (13), as follows:

$$\hat{\mathbf{o}}_t^S = \hat{\mathbf{o}}_t^1 + \sum_{s=1}^{S-1} \hat{\mathbf{r}}_t^s \in \mathbb{R}^q. \tag{14}$$

As discussed by He *et al.* [9], if the added layers can be constructed as identity mappings, a deeper model should have a training error no greater than its shallower counterpart. In the proposed model, if the $(s-1)$-th residual compensation layer has already achieved perfect prediction such that $\hat{\mathbf{o}}_t^{s-1} = \mathbf{o}_t$, the following s-th residual compensation layer should be an identity mapping. The identified mapping can be easily constructed. Because, from Eq. (11), the ELM target feature vectors to $\mathbf{y}_t = \mathbf{r}_t^{s-1} = \mathbf{0}$, and the optimal weight matrix is obtained as $\boldsymbol{\beta}^* = \mathbf{0}$, which results in $\hat{\mathbf{r}}_t^s = \hat{\mathbf{y}}_t = \mathbf{0}$.

The dilation pattern in the proposed method is the same as that used in WaveNet, in that the dilation is doubled for every layer up to a certain limit and the pattern is then repeated [11]:

$$n_s = 2^{(s-1)\mathrm{mod}W}, \tag{15}$$

where W denotes the repeat period. The receptive field of the multilayer network stacking S layers is then given by

$$RF\,(S, W) = 1 + \sum_{s=1}^{S} n_s. \tag{16}$$

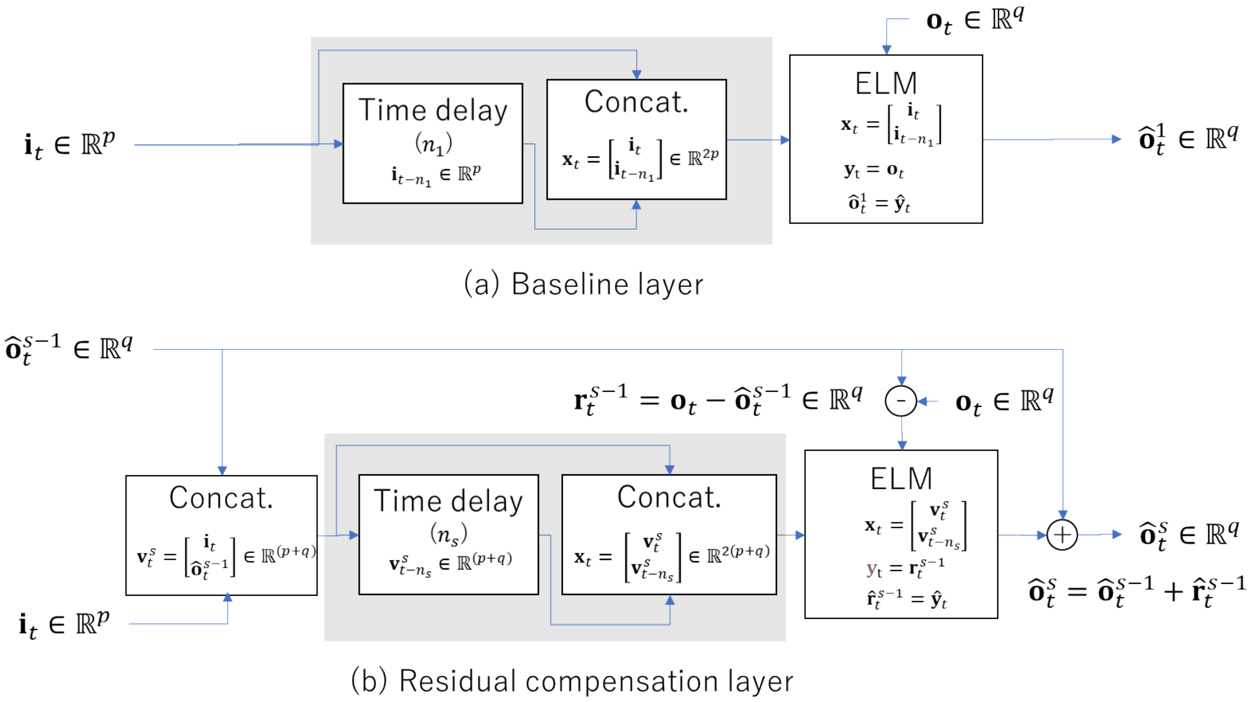
For instance, in the case of $S = W = 4$, the dilation pattern is represented by the time delay parameters, $(n_s)_{s=1,\cdots,4} = (1, 2, 4, 8)$. The prediction was equivalently conducted according to the block diagram shown in Fig. 3. This multilayer network calculates the prediction of the final layer, $\hat{\mathbf{o}}_t^4$, based on the original input feature vectors in a range from time t to $t-15$. Thus, the receptive field is 16, as given by Eq. (16).

IV. EXPERIMENTS

The proposed DRLDCC-ELM has the following characteristics:

A. Several residual compensation layers are iteratively stacked to remodel the uncaptured prediction error for the previous layer.
B. The predicted target features for the previous layer are fed into the next residual-compensation layer following concatenation with the original input features.
C. Each layer is subjected to a dilated causal convolution, where the dilation step size is different for each layer.

Characteristic A is shared by the RC-ELM, DRC-ELM, and DRLDCC-ELM. Characteristic B is lacking in the RC-ELM because the predicted target features for the previous layer are not fed into the next layer. Characteristic C is lacking in both the RC-ELM and DRC-ELM because some of the same input features are repeatedly fed into each layer. Table 1 summarizes the characteristics of each method, where an alternative RC-ELM is detailed in subsection B-2).


$\mathbf{o}_t \in \mathbb{R}^q$
$\mathbf{i}_t \in \mathbb{R}^p$
Time delay
(n_1)
$\mathbf{i}_{t-n_1} \in \mathbb{R}^p$
Concat.
$\mathbf{x}_t = \begin{bmatrix} \mathbf{i}_t \\ \mathbf{i}_{t-n_1} \end{bmatrix} \in \mathbb{R}^{2p}$
ELM
$\mathbf{x}_t = \begin{bmatrix} \mathbf{i}_t \\ \mathbf{i}_{t-n_1} \end{bmatrix}$
$\mathbf{y}_t = \mathbf{o}_t$
$\hat{\mathbf{o}}_t^1 = \hat{\mathbf{y}}_t$
$\hat{\mathbf{o}}_t^1 \in \mathbb{R}^q$
$\hat{\mathbf{o}}_t^{s-1} \in \mathbb{R}^q$
$\mathbf{r}_t^{s-1} = \mathbf{o}_t - \hat{\mathbf{o}}_t^{s-1} \in \mathbb{R}^q$
-
$\mathbf{o}_t \in \mathbb{R}^q$
Concat.
$\mathbf{v}_t^s = \begin{bmatrix} \mathbf{i}_t \\ \hat{\mathbf{o}}_t^{s-1} \end{bmatrix} \in \mathbb{R}^{(p+q)}$
$\mathbf{i}_t \in \mathbb{R}^p$
Time delay
(n_s)
$\mathbf{v}_{t-n_s}^s \in \mathbb{R}^{(p+q)}$
Concat.
$\mathbf{x}_t = \begin{bmatrix} \mathbf{v}_t^s \\ \mathbf{v}_{t-n_s}^s \end{bmatrix} \in \mathbb{R}^{2(p+q)}$
ELM
$\mathbf{x}_t = \begin{bmatrix} \mathbf{v}_t^s \\ \mathbf{v}_{t-n_s}^s \end{bmatrix}$
$\mathbf{y}_t = \mathbf{r}_t^{s-1}$
$\hat{\mathbf{r}}_t^{s-1} = \hat{\mathbf{y}}_t$
+
$\hat{\mathbf{o}}_t^s \in \mathbb{R}^q$
$\hat{\mathbf{o}}_t^s = \hat{\mathbf{o}}_t^{s-1} + \hat{\mathbf{r}}_t^{s-1}$

(a) Baseline layer

(b) Residual compensation layer

FIGURE 2. Network architecture of the proposed DRLDCC-ELM. (a) Baseline layer. (b) Residual compensation layer.

TABLE 1. Summary of the characteristics of each method.

Method	Characteristics		
	A	B	C
RC-ELM	○		
Alternative RC-ELM	○		○
DRC-ELM	○	○	
DRLDCC-ELM	○	○	○

Characteristics B and C are very important for the proposed DRLDCC-ELM. In the following experiments, the effectiveness of these characteristics was evaluated.

A. DATASET

The task we chose was acoustic scene classification of daily activities in a home environment, and we conducted experiments using a dataset named "*SINS*" that contains recordings of the activities of a person at home obtained using an acoustic sensor network. The dataset was collected in a vacation home consisting of five different rooms: a combined living room and kitchen, a bathroom, a toilet, a bedroom, and a hall. Thirteen sensor nodes, each containing four microphones, were distributed uniformly around the five rooms [12]. Each audio channel was sampled at a rate of 16 kHz with a bit depth of 12. In our experiments, we selected the recordings of living room activities because they were the most varied. Nine kinds of activities were included in our analysis: making a phone call, cooking, dishwashing, eating, visiting, watching TV, working, vacuum cleaning, and the absence of any activity. Other activities were excluded. We used the monaural channel #1 audio signal extracted from the four audio channels recorded by the microphone array of node 7 situated in the living room. Overall, the dataset was highly variable, reflecting an imbalance of activities occur in daily life. In order to balance the dataset, we therefore extracted data segments with a duration of one minute that included only one activity, and, for each activity, we concatenated the segments to generate training and testing datasets with durations of 30 minutes and 15 minutes, respectively. Because there were nine kinds of activity, the total durations of these datasets were 270 min and 135 min, respectively.

B. METHODS AND RESULTS

1) PROPOSED DRLDCC-ELM

During front-end processing, mel-spectrogram features were extracted as original input features, $\boldsymbol{i}_t \in \mathbb{R}^p$, from the monaural audio signals pre-emphasized with a difference filter of $1 - 0.97\, z^{-1}$. The size and shift of the analysis frame were set to 30 ms and 20 ms, respectively. The dimensions of the mel-spectrogram feature vector were $p = 40$. In the proposed DRLDCC-ELM, the concatenated feature vectors given by Eq. (6) are fed into the baseline layer of ELM. Thus, for the baseline layer, the number of nodes in the ELM input layer was set to $M = 2p = 80$. In this experiment, a nine-class classification task was conducted, so the original target feature vectors, $\boldsymbol{o}_t \in \mathbb{R}^q$, were one-hot vectors with dimensions of nine. The number of nodes in the ELM output layer was set to $K = q = 9$. For the residual-compensation layer, the input

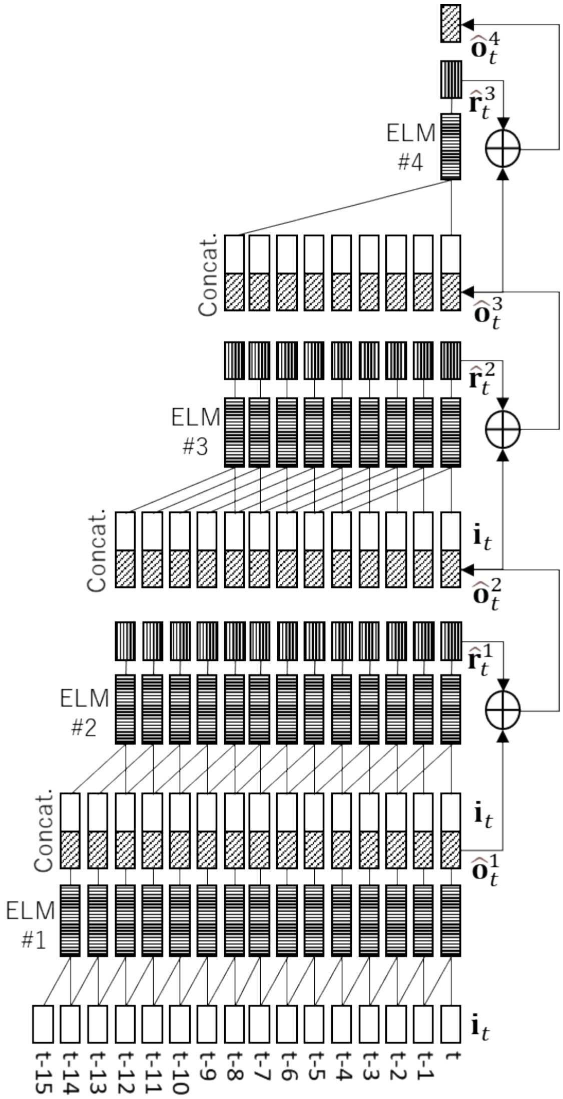


FIGURE 3. The equivalent block diagram and receptive field of the DRLDCC-ELM that was stacked under the dilation pattern of (1,2,4,8).

TABLE 2. Experimental results for the proposed DRLDCC-ELM.

Number of nodes in hidden layer (L)	Number of stacked layers (S)								
	10			11			12		
	Receptive fields (RF)								
	1024			2048			4096		
	Number of learnable parameters	Macro averaged F1-score		Number of learnable parameters	Macro averaged F1-score		Number of learnable parameters	Macro averaged F1-score	
		Avg	Std Dev		Avg	Std Dev		Avg	Std Dev
1,000	90,000	80.83%	0.26%	99,000	83.94%	0.37%	108,000	88.03%	0.63%
2,000	180,000	83.35%	0.67%	198,000	87.30%	0.60%	216,000	91.28%	0.56%
3,000	270,000	84.53%	0.41%	297,000	88.27%	0.45%	324,000	91.92%	0.57%
4,000	360,000	85.11%	0.38%	396,000	88.87%	0.45%	432,000	92.72%	0.39%
5,000	450,000	85.89%	0.27%	495,000	89.73%	0.22%	540,000	93.47%	0.27%
6,000	540,000	86.37%	0.41%	594,000	90.28%	0.41%	648,000	93.84%	0.41%
7,000	630,000	86.68%	0.53%	693,000	90.51%	0.43%	756,000	94.14%	0.35%
8,000	720,000	86.91%	0.18%	792,000	90.74%	0.12%	864,000	94.29%	0.29%
9,000	810,000	87.19%	0.21%	891,000	91.01%	0.29%	972,000	94.59%	0.39%
10,000	900,000	87.54%	0.20%	990,000	91.36%	0.18%	1,080,000	94.56%	0.15%

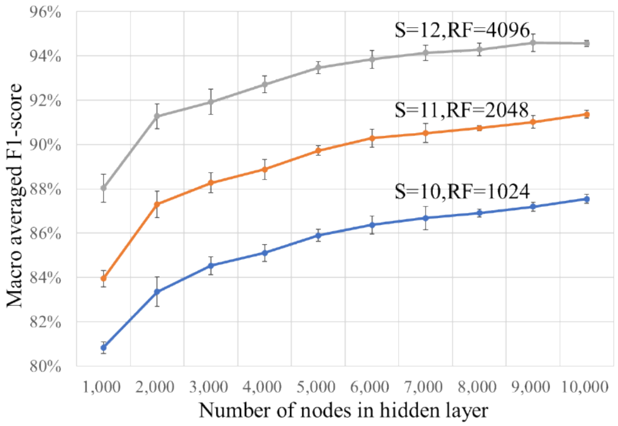


FIGURE 4. F1-scores of the proposed DRLDCC-ELM show three graphs in which the number of stacked layers ranges from 10 to 12. The horizontal axis represents the number of hidden layer nodes.

feature vectors given by Eq. (10) were fed into the residual-compensation layer ELM, so the number of nodes in the ELM input layer was set to $M = 2(p+q) = 98$. The nonlinear activation functions and other user-defined values were the same for all ELMs. A rectified linear unit (ReLU) function was adopted as the nonlinear activation function in Eq. (1). For the number of nodes in the hidden layer, we used ten values from $L = 1000$ to 10000, with a step size of 1000. The weight of the prediction error term in Eq. (4) was set to $C = 1.0$. The number of stacked layers was set to $S = 10, 11, 12$. The repeat period for the dilation pattern given in Eq. (15) was set to $W = 12$. The receptive fields for $S = 10, 11, 12$ are given by Eq. (16) as $RF = 1024, 2048, 4096$, respectively. The learning model was iterated five times for each condition.

Table 2 shows the experimental results for the proposed DRLDCC-ELM, where the average and standard deviation of the average F1 scores evaluated from five trials are presented. The table presents the total number of learnable parameters, which is given by $K \times L \times S$, for each combination of the number of stacked layers, S, and the number of nodes in the hidden layer, L. The DRLDCC-ELM predicts the target output features for the same frame-shift period as the original input features. We evaluated the F1 score for each class by comparing the predictions with the correct labels for each frame. We then adopted the average F1 score as a metric for multiclass classification. These results are summarized in Table 2, and the graphs are shown in Fig. 4. The average F1 score increased almost monotonically as the number of nodes in the hidden layer and/or the number of stacked layers increased.

2) COMPARISON WITH RC-ELM

In order to evaluate the effectiveness of the aforementioned characteristic B, instead of comparing the DRLDCC-ELM directly with the original RC-ELM lacking characteristics B and C, we conducted experiments using an alternative method lacking only characteristic B. In this alternative method, the predicted target features of the previous layer were not fed into the next residual-compensation layer, but only the

TABLE 3. Experimental results for the alternative RC-ELM method and the extended DRC-ELM.

Number of nodes in hidden layer (L)	Compared methods					
	Alternative RC-ELM			DRC-ELM		
	Number of stacked layers (S)					
	10			10		
	Receptive fields (RF)					
	1024			1031		
	Number of learnable weights	Macro averaged F1-score		Number of learnable weights	Macro averaged F1-score	
		Avg	Std Dev		Avg	Std Dev
1,000	90,000	59.62%	0.20%	90,000	72.46%	0.36%
2,000	180,000	61.34%	0.17%	180,000	76.91%	0.31%
3,000	270,000	62.39%	0.22%	270,000	78.17%	0.12%
4,000	360,000	63.38%	0.06%	360,000	79.31%	0.15%
5,000	450,000	63.73%	0.09%	450,000	79.75%	0.27%
6,000	540,000	64.36%	0.07%	540,000	80.58%	0.15%
7,000	630,000	64.87%	0.14%	630,000	81.27%	0.24%
8,000	720,000	65.22%	0.13%	720,000	81.35%	0.11%
9,000	810,000	65.52%	0.05%	810,000	81.87%	0.14%
10,000	900,000	65.84%	0.12%	900,000	82.08%	0.21%

original input features were fed into. The alternative method is thus lacking characteristic B, as in the case of the original RC-ELM. On the other hand, the alternative method executes the dilation convolution the same as in DRLDCC-ELM. Because of this, the alternative method has characteristic C. The feature vector, given in Eq. (9) consists of only the original input feature vector, $\mathbf{i}_t \in \mathbb{R}^p$, therefore, the ELM input feature vector in the residual-compensation layer given by Eq. (10) is the same as that in the baseline layer, as given by Eq. (6). The total number of learnable parameters is the same as that for the DRLDCC-ELM. In this experiment, we considered only the case where the number of stacked layers was $S = 10$. The learning model was iterated five times for each condition. The results are presented in Table 3, and a graph of the results is shown in Fig. 5. The average F1 score for the DRLDCC-ELM is 21.81% higher than that of the alternative RC-ELM method.

3) COMPARISON WITH DRC-ELM

Next, we evaluated the effectiveness of characteristic C by conducting experiments using DRC-ELM, which lacks this characteristic. Instead of the dilation convolution adopted in the DRLDCC-ELM, $(N + 1)$ successive original input features were fed into the baseline layer and were concatenated with the predicted original target feature vector for the previous layer, and then repeatedly fed into the residual-compensation layers. Figs. 6(a) and 6(b) depict the baseline layer and residual-compensation layer for the DRC-ELM, respectively. As shown, the DRC-ELM used in this experiment can be constructed by embedding a successive feature sampling block instead of the feature preprocessing block used in the DRLDCC-ELM, as depicted in Fig. 2. Owing to adopting successive feature sampling blocks, the DRC-ELM used in this experiment is slightly extended from the original DRC-ELM such that each residual compensation layer can utilize the previous layer's past original target feature predictions to predict the current original target feature. When $(N + 1)$ successive original input features, including the current time, are fed into the ELMs for both the baseline layer and all the prediction-residual layers, each sample time delay block needs to be concatenated N times. (The maximum delay time is thus N.) All of the time-delayed features as well as the current time feature are concatenated and then fed into each ELM. The receptive field for this type of multilayer network is given by $RF\,(S, N) = 1 + SN$. To ensure that the experimental conditions were the same as those for the DRLDCC-ELM, where the number of stacked layers was $S = 10$ $(RF = 1024)$, we set the number of stacked layers and the maximum delay time to $S = 10$ and $N = 103$, respectively. In this case, the receptive field for the DRC-ELM is $RF = 1031$, which is slightly larger than that for DRLDCC-ELM and may be advantageous for DRC-ELM. The learning model was iterated five times for each condition. The results are presented in Table 3 and the graphs are shown in Fig. 5. It can be seen that the average F1 score for DRLDCC-ELM is higher than that for DRC-ELM by 6.07%.

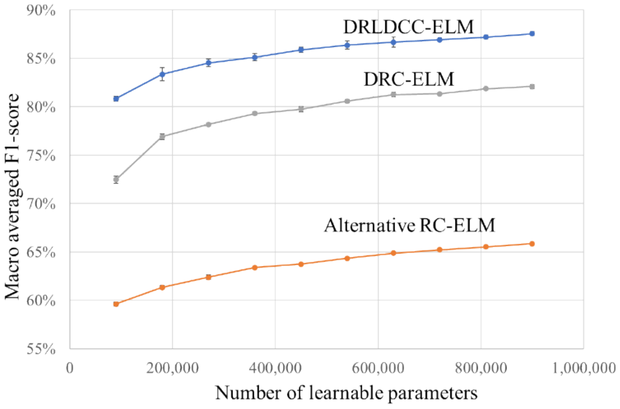


FIGURE 5. Comparisons of F1-scores of the proposed DRLDCC-ELM with those of the alternative RC-ELM method and the extended DRC-ELM. The horizontal axis shows the number of learnable parameters.

4) COMPARISON WITH CNN-BASED METHOD

Finally, we compared the DRLDCC-ELM with convolutional neural network (CNN)-based models. We constructed CNN-based networks by modifying a previously proposed network architecture [13]. The network architectures used in the following experiments were constructed by combining the element blocks presented in Table 4. The two-dimensional features to be fed into the input block consisted of the mel-spectrogram (40 dimensions) of 1024 successive frames; thus, the total number of dimensions was 40 × 1024. Four types of network architectures were constructed by inserting several middle blocks between the input and output blocks, as shown in Table 5. An Adam optimization algorithm was adopted for stochastic gradient descent for training deep learning models, and the learning rate was set to 0.0001. The learning model was iterated five times for each condition.

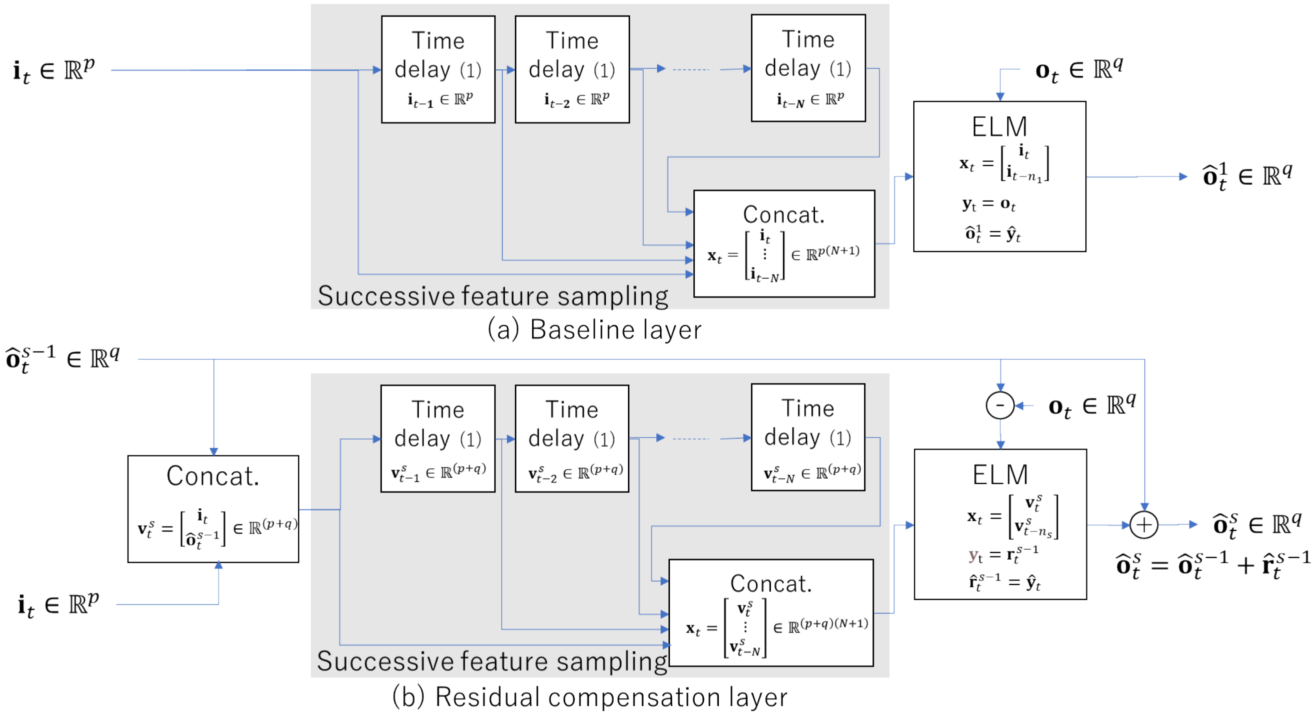

$\mathbf{i}_t \in \mathbb{R}^p$
Time delay (1) $\mathbf{i}_{t-1} \in \mathbb{R}^p$
Time delay (1) $\mathbf{i}_{t-2} \in \mathbb{R}^p$
Time delay (1) $\mathbf{i}_{t-N} \in \mathbb{R}^p$
$\mathbf{o}_t \in \mathbb{R}^q$
ELM
$\mathbf{x}_t = \begin{bmatrix}\mathbf{i}_t\\ \mathbf{i}_{t-n_1}\end{bmatrix}$
$\mathbf{y}_t = \mathbf{o}_t$
$\hat{\mathbf{o}}_t^1 = \hat{\mathbf{y}}_t$
$\hat{\mathbf{o}}_t^1 \in \mathbb{R}^q$
Concat.
$\mathbf{x}_t = \begin{bmatrix}\mathbf{i}_t\\ \vdots\\ \mathbf{i}_{t-N}\end{bmatrix} \in \mathbb{R}^{p(N+1)}$
Successive feature sampling
(a) Baseline layer
$\hat{\mathbf{o}}_t^{s-1} \in \mathbb{R}^q$
Concat.
$\mathbf{v}_t^s = \begin{bmatrix}\mathbf{i}_t\\ \hat{\mathbf{o}}_t^{s-1}\end{bmatrix} \in \mathbb{R}^{(p+q)}$
$\mathbf{i}_t \in \mathbb{R}^p$
Time delay (1) $\mathbf{v}_{t-1}^s \in \mathbb{R}^{(p+q)}$
Time delay (1) $\mathbf{v}_{t-2}^s \in \mathbb{R}^{(p+q)}$
Time delay (1) $\mathbf{v}_{t-N}^s \in \mathbb{R}^{(p+q)}$
$\mathbf{o}_t \in \mathbb{R}^q$
ELM
$\mathbf{x}_t = \begin{bmatrix}\mathbf{v}_t^s\\ \mathbf{v}_{t-n_s}^s\end{bmatrix}$
$\mathbf{y}_t = \mathbf{r}_t^{s-1}$
$\hat{\mathbf{r}}_t^{s-1} = \hat{\mathbf{y}}_t$
$\hat{\mathbf{o}}_t^s \in \mathbb{R}^q$
$\hat{\mathbf{o}}_t^s = \hat{\mathbf{o}}_t^{s-1} + \hat{\mathbf{r}}_t^{s-1}$
Concat.
$\mathbf{x}_t = \begin{bmatrix}\mathbf{v}_t^s\\ \vdots\\ \mathbf{v}_{t-N}^s\end{bmatrix} \in \mathbb{R}^{(p+q)(N+1)}$
Successive feature sampling
(b) Residual compensation layer

FIGURE 6. Network architecture of the extended DRC-ELM. (a) Baseline layer. (b) Residual compensation layer.

TABLE 4. The element blocks used for the network architecture of CNN-based model.

	Layer	Description
Input block	input	40x1024 (mel-spectrogram)
	convolution (2D)	64 filters, kernel=(5,5), stride=(2,2), padding=(2,2)
	batch normalization	
	ReLU	
	convolution (2D)	64 filters, kernel=(3,3), stride=(1,1), padding=(1,1)
	batch normalization	
	ReLU	
	max pooling (2D)	kernel=(3,2), stride=(1,2), padding=(1,0), dilation=(1,1)
	dropout	p=0.3
	convolution (2D)	128 filters, kernel=(3, 3), stride=(1, 1), padding=(1, 1)
	batch normalization	
	ReLU	
Middle block 1	max pooling (2D)	kernel=(3,2), stride=(1,2), padding=(1,0), dilation=(1,1)
	dropout	p=0.3
Middle block 2	convolution (2D)	F filters, kernel=(3, 3), stride=(1, 1), padding=(1, 1)
	batch normalization	
	ReLU	
	dropout	p=0.3
Middle block 3	convolution (2D)	256 filters, kernel=(3, 3), stride=(1, 1), padding=(1, 1)
	batch normalization	
	ReLU	
	dropout	p=0.3
Output block	convolution (2D)	32 filters, kernel=(1,1), stride=(1,1)
	batch normalization	
	ReLU	
	adaptive average pool	output size=(1,1)
	flatten	
	linear	out_features=16
	ReLU	
	linear	out_features=9
	softmax	

The average and standard deviation of the F1 scores evaluated from the five trials are presented in Table 6, together with the number of learnable parameters. Fig. 7 shows a graph of the results. When the number of learnable parameters was less than approximately 265k, the CNN-based model outperformed the DRLDCC-ELM model. When the number of learnable parameters was between 265k and 565k, the DRLDCC-ELM performance was comparable to that of the CNN-based models. For a number larger than approximately 565k, the F1 scores for the CNN-based models decreased slightly. However, the F1 scores for the DRLDCC-ELM increased constantly. Thus, we confirmed that the generalization capability of the DRLDCC-ELM is superior to that of the CNN-based models when the number of parameters is large.

TABLE 5. Four types of network architecture for the CNN-based models.

CNN-1	CNN-2	CNN-3	CNN-4
input block	input block	input block	input block
middle-1 block	middle-2 block	middle-2 block	middle-2 block
output block	output block	middle-3 block	middle-3 block
		output block	middle-3 block
			output block

TABLE 6. Experimental results for the CNN-based models.

CNN-models	Number of learnable parameters	Macro averaged F1-score	
		Avg	Std Dev
CNN-1	117,833	84.11%	0.54%
CNN-2	265,673	84.90%	0.40%
CNN-3	565,449	86.35%	0.68%
CNN-4	1,156,041	86.00%	0.63%

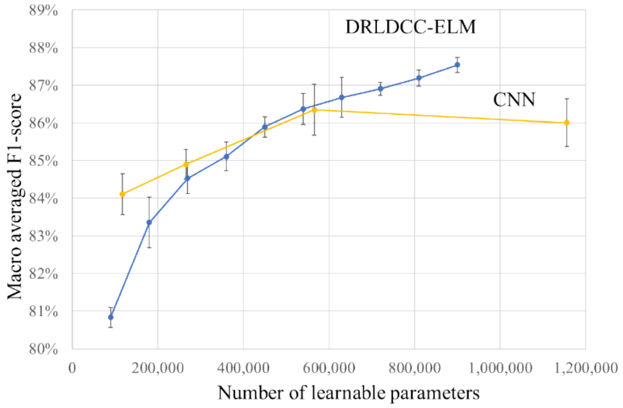


FIGURE 7. Comparisons of F1-scores of the proposed DRLDCC-ELM with those of the CNN-based models. The horizontal axis shows the number of learnable parameters.

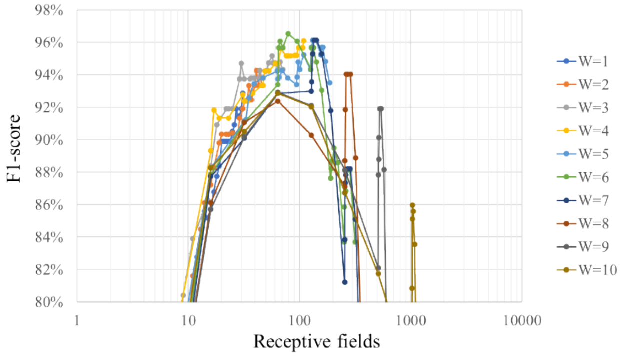


FIGURE 8. Examples of sound event detection results. The horizontal axis shows the number of frames in the receptive fields.

V. HYPERPARAMETER OPTIMIZATION

The proposed DRLDCC-ELM has the following hyperparameters:

- The number, L, of nodes in the hidden layers.
- The weight C, for the prediction error term in Eq. (4).
- The number, S, of stacked layers in Eq. (16).
- The repeat period W, of the dilation pattern in Eq. (16).

In the present study, the number of nodes in the hidden layers ranged from 1000 to 10000. The weight for the prediction error term was set to an empirically determined value of 1.0. The number of stacked layers and the repeat period for the dilation pattern both have a significant impact on the performance of the resulting network. The grid search method has traditionally been used for hyperparameter optimization. This technique investigates all combinations of hyperparameters by evaluating the performance of the resulting networks. However, using this method with the DRLDCC-ELM, if there are a large number of hyperparameter combinations, intensive calculations will be required irrespective of the main advantage of the ELM; that is, it can optimize the network weights by solving a single standard linear least-squares problem. To avoid this situation, we introduce a fundamental concept to effectively search for the semi-optimal hyperparameters S and W in Eq. (16) from the training data for sound event detection.

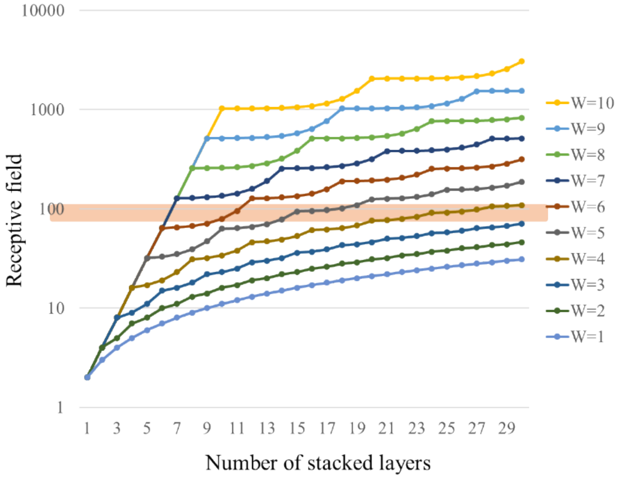


FIGURE 9. Receptive fields, (S, W), calculated in conditions that S was set to a number ranging from 1 to 30 and W was set to a number ranging from 1 to 10.

In this approach, the DRLDCC-ELM was trained to detect the existence of only one target sound event, frame by frame, from a mel-spectrogram time series. First, we conducted a grid search. The models were trained on all combinations of hyperparameters, where S was 1–30 and W was 1–10. Therefore, the number of hyperparameter combinations was 300, and the number of trained models was also 300. We evaluated the F1 score of each model tested the models on the test data. Fig. 8 shows the evaluated F1 scores for all the trained models, where scores obtained with a fixed value of W are connected by a line, and the horizontal axis represents the receptive field, $RF(S, W)$, in Eq. (16). Irrespective of the value of W, all the lines are convex curves, and the maximum F1 scores are obtained for receptive fields of approximately 100 frames.

We then investigated the duration of the target sound events in the training data. The statistics obtained are as follows: The minimum duration was 46 frames, the maximum duration was 456 frames, and the average duration was 108 frames with a standard deviation of 66 frames. Therefore, the maximum F1 scores tended to be obtained with models trained with the same receptive field as the average duration of the target sound event in the training data. This implies that only models whose receptive field is around the average duration of the target sound event in the training data should be considered as candidates for the optimal model. Fig. 9 shows the receptive fields calculated for all combinations of S and W. Receptive fields obtained with a fixed value of W are connected by a line, and the horizontal axis represents the number of stacked layers, S. The vertical axis represents the receptive field on a logarithmic scale. In the graph, one dot represents one combination of S and W, that is, one model. For the grid search method, all dots (models) in the graph were trained and evaluated. However, if we focus only on models with receptive fields that are around the average duration of the target sound event, we can drastically reduce the number of models to be considered. In the graph, the models depicted in the orange area can be regarded as candidates for optimal models.

VI. CONCLUSION

We proposed a modified architecture based on deep residual-compensation ELM to effectively utilize past input features over a larger area by introducing a dilated causal convolution ELM. In the proposed DRLDCC-ELM, several residual compensation layers are iteratively stacked to remodel the uncaptured prediction errors in the previous layer, the predicted target features of the previous layer are then concatenated with the original input features and fed into the next residual-compensation layer, and each layer has a dilated causal convolution with a different dilation step size. The experimental results confirmed that the latter two features are necessary for the DRLDCC-ELM to outperform the previously proposed RC-ELM and DRC-ELM. We also found that the generalization capability of the DRLDCC-ELM was superior to that of the CNN-based models, especially for larger numbers of parameters.

REFERENCES

[1] S. Scardapane and D. Wang, "Randomness in neural networks: An overview," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 7, no. 2, 2017, Art. no. e1200, doi: 10.1002/widm.1200.

[2] Y.-H. Pao, G.-H. Park, and D. Sobajic, "Learning and generalization characteristics of the random vector functional-link net," *Neurocomputing*, vol. 6, no. 2, 1994, Art. no. 163180.

[3] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012, doi: 10.1109/TSMCB.2011.2168604.

[4] J. Wang, S. Lu, S.-H. Wang, and Y.-D. Zhang, "A review on extreme learning machine," *Multimedia Tools Appl.*, pp. 1–50, May 2021, doi: 10.1007/s11042-021-11007-7.

[5] L. L. C. Kasun, H. Zhou, G. B. Huang, and C. M. Vong, "Representational learning with ELMs for big data," *IEEE Intell. Syst.*, vol. 28, no. 6, pp. 31–34, Nov. 2013.

[6] J. Zhang, W. Xiao, Y. Li, and S. Zhang, "Residual compensation extreme learning machine for regression," *Neurocomputing*, vol. 311, pp. 126–136, Oct. 2018, doi: 10.1016/j.neucom.2018.05.057.

[7] S. Zhang, Z. Liu, X. Huang, and W. Xiao, "A modified residual extreme learning machine algorithm and its application," *IEEE Access*, vol. 6, pp. 62215–62223, 2018, doi: 10.1109/ACCESS.2018.2876360.

[8] Y. Chen, X. Xie, T. Zhang, J. Bai, and M. Hou, "A deep residual compensation extreme learning machine and applications," *J. Forecasting*, vol. 39, no. 6, pp. 986–999, Sep. 2020, doi: 10.1002/for.2663.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[10] G.-B. Huang, Z. Bai, L. L. C. Kasun, and C. M. Vong, "Local receptive fields based extreme learning machine," *IEEE Comput. Intell. Mag.*, vol. 10, no. 2, pp. 18–29, May 2015, doi: 10.1109/MCI.2015.2405316.

[11] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," in *Proc. 9th ISCA Speech Synth. Workshop*, 2016, pp. 1–15.

[12] G. Dekkers, S. Lauwereins, B. Thoen, M. W. Adhana, H. Brouckxon, T. V. Waterschoot, B. Vanrumste, M. Verhelst, and P. Karsmakers, "The SINS database for detection of daily activities in a home environment using an acoustic sensor network," in *Proc. Detection Classification-Acoustic Scenes Events Workshop (DCASE)*, Nov. 2017, pp. 32–36.

[13] M. Dorfer and G. Widmer, "Training general-purpose audio tagging networks with noisy labels and iterative self-verification," in *Proc. DCASE Challenge*, 2018, pp. 178–182.

[14] X. Peng, P. Lin, T. Zhang, and J. Wang, "Extreme learning machine-based classification of ADHD using brain structural MRI data," *PLoS ONE*, vol. 8, no. 11, Nov. 2013, Art. no. e79476.

[15] M. Termenon, M. Graña, A. Barrós-Loscertales, and C. Ávila, "Extreme learning machines for feature selection and classification of cocaine dependent patients on structural MRI data," *Neural Process. Lett.*, vol. 38, no. 3, pp. 375–387, Dec. 2013.

[16] M. N. I. Qureshi, B. Min, H. J. Jo, and B. Lee, "Multiclass classification for the differential diagnosis on the ADHD subtypes using recursive feature elimination and hierarchical extreme learning machine: Structural MRI study," *PLoS ONE*, vol. 11, no. 8, Aug. 2016, Art. no. e0160697.

[17] M. Termenon, M. Graña, A. Savio, A. Akusok, Y. Miche, K.-M. Björk, and A. Lendasse, "Brain MRI morphological patterns extraction tool based on extreme learning machine and majority vote classification," *Neurocomputing*, vol. 174, pp. 344–351, Jan. 2016.

[18] S. Lu, Z. Lu, J. Yang, M. Yang, and S. Wang, "A pathological brain detection system based on kernel based ELM," *Multimedia Tools Appl.*, vol. 77, no. 3, pp. 3715–3728, Feb. 2018.

[19] R. K. Lama, J. Gwak, J. S. Park, and S. W. Lee, "Diagnosis of Alzheimer's disease based on structural MRI images using a regularized extreme learning machine and PCA features," *J. Healthcare Eng.*, vol. 2017, Jun. 2017, Art. no. 5485080.

[20] M. N. I. Qureshi, J. Oh, D. Cho, H. J. Jo, and B. Lee, "Multimodal discrimination of schizophrenia using hybrid weighted feature concatenation of brain functional connectivity and anatomical features with an extreme learning machine," *Frontiers Neuroinform.*, vol. 11, p. 59, Sep. 2017.

[21] Y.-D. Zhang, G. Zhao, J. Sun, X. Wu, Z.-H. Wang, H.-M. Liu, V. V. Govindaraj, T. Zhan, and J. Li, "Smart pathological brain detection by synthetic minority oversampling technique, extreme learning machine, and Jaya algorithm," *Multimedia Tools Appl.*, vol. 77, no. 17, pp. 22629–22648, Sep. 2018.

[22] D. R. Nayak, D. Das, R. Dash, S. Majhi, and B. Majhi, "Deep extreme learning machine with leaky rectified linear unit for multiclass classification of pathological brain images," *Multimedia Tools Appl.*, vol. 79, nos. 21–22, pp. 15381–15396, Jun. 2020.

[23] D. T. Nguyen, S. Ryu, M. N. I. Qureshi, M. Choi, K. H. Lee, and B. Lee, "Hybrid multivariate pattern analysis combined with extreme learning machine for Alzheimer's dementia diagnosis using multi-measure rs-fMRI spatial patterns," *PLoS ONE*, vol. 14, no. 2, Feb. 2019, Art. no. e0212582.

[24] A. Gumaei, M. M. Hassan, M. R. Hassan, A. Alelaiwi, and G. Fortino, "A hybrid feature extraction method with regularized extreme learning machine for brain tumor classification," *IEEE Access*, vol. 7, pp. 36266–36273, 2019.

[25] W. Huang, Z. M. Tan, Z. Lin, G. Huang, J. Zhou, C. K. Chui, Y. Su, and S. Chang, "A semi-automatic approach to the segmentation of liver parenchyma from 3D CT images with extreme learning machine," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2012, pp. 28–33.

[26] G. L. B. Ramalho, P. P. Rebouças Filho, F. N. S. D. Medeiros, and P. C. Cortez, "Lung disease detection using feature extraction and extreme learning machine," *Revista Brasileira Engenharia Biomédica*, vol. 30, no. 3, pp. 207–214, Sep. 2014.

[27] J. Xia, "Ultrasound-based differentiation of malignant and benign thyroid nodules: An extreme learning machine approach," *Comput. Methods Programs Biomed.*, vol. 147, pp. 37–49, Oct. 2017.

[28] M. Niu, J. Zhang, Y. Li, C. Wang, Z. Liu, H. Ding, Q. Zou, and Q. Ma, "CirRNAPL: A web server for the identification of circRNA based on extreme learning machine," *Comput. Struct. Biotechnol. J.*, vol. 18, pp. 834–842, Apr. 2020, doi: 10.1016/j.csbj.2020.03.028.

[29] J. L. Song, W. Hu, and R. Zhang, "Automated detection of epileptic EEGs using a novel fusion feature and extreme learning machine," *Neurocomputing*, vol. 175, pp. 383–391, Jan. 2016.

[30] Y. Qin, M. Li, G. De, L. Huang, S. Yang, Q. Tan, Z. Tan, and F. Zhou, "Research on green management effect evaluation of power generation enterprises in China based on dynamic hesitation and improved extreme learning machine," *Processes*, vol. 7, no. 7, p. 474, Jul. 2019.

[31] S. Singh, M. Pareek, A. Changotra, S. Banerjee, B. Bhaskararao, P. Balamurugan, and R. B. Sunoj, "A unified machine-learning protocol for asymmetric catalysis as a proof of concept demonstration using asymmetric hydrogenation," *Proc. Nat. Acad. Sci. USA*, vol. 117, no. 3, pp. 1339–1345, Jan. 2020.

[32] H. Lei, Y. Wen, Z. You, A. Elazab, E.-L. Tan, Y. Zhao, and B. Lei, "Protein–protein interactions prediction via multimodal deep polynomial network and regularized extreme learning machine," *IEEE J. Biomed. Health Informat.*, vol. 23, no. 3, pp. 1290–1303, May 2019.

[33] I. Landa-Torres, E. G. Ortiz-Garcia, S. Salcedo-Sanz, M. J. Segovia-Vargas, S. Gil-Lopez, M. Miranda, J. M. Leiva-Murillo, and J. D. Ser, "Evaluating the internationalization success of companies through a hybrid grouping harmony search—Extreme learning machine approach," *IEEE J. Sel. Topics Signal Process.*, vol. 6, no. 4, pp. 388–398, Aug. 2012.

[34] D. Cogoljević, M. Alizamir, I. Piljan, T. Piljan, K. Prljić, and S. Zimonjić, "A machine learning approach for predicting the relationship between energy resources and economic development," *Phys. A, Stat. Mech. Appl.*, vol. 495, pp. 211–214, Apr. 2018.

[35] D. Marković, D. Petković, V. Nikolić, M. Milovančević, and B. Petković, "Soft computing prediction of economic growth based in science and technology factors," *Phys. A, Stat. Mech. Appl.*, vol. 465, pp. 217–220, Jan. 2017.

[36] V. Marjanović, M. Milovančević, and I. Mladenović, "Prediction of GDP growth rate based on carbon dioxide (CO2) emissions," *J. CO2 Utilization*, vol. 16, pp. 212–217, Dec. 2016.

[37] L. Milačić, S. Jović, T. Vujović, and J. Miljković, "Application of artificial neural network with extreme learning machine for economic growth estimation," *Phys. A, Stat. Mech. Appl.*, vol. 465, pp. 285–288, Jan. 2017.

[38] G. Rakic, D. Milenkovic, S. Vujovic, T. Vujovic, and S. Jović, "Information system for e-GDP based on computational intelligence approach," *Phys. A, Stat. Mech. Appl.*, vol. 513, pp. 418–423, Jan. 2019.

[39] J. Sánchez-Oro, A. Duarte, and S. Salcedo-Sanz, "Robust total energy demand estimation with a hybrid variable neighborhood search—Extreme learning machine algorithm," *Energy Convers. Manage.*, vol. 123, pp. 445–452, Sep. 2016.

[40] S. Z. H. Shoumo, M. I. M. Dhruba, S. Hossain, N. H. Ghani, H. Arif, and S. Islam, "Application of machine learning in credit risk assessment: A prelude to smart banking," in *Proc. IEEE Region Conf. (TENCON)*, Oct. 2019, pp. 2023–2028.

[41] Z.-L. Sun, K. M. Ng, J. Soszynska-Budny, and M. S. Habibullah, "Application of the LP-ELM model on transportation system lifetime optimization," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1484–1494, Dec. 2011.

[42] Y. Zeng, X. Xu, D. Shen, Y. Fang, and Z. Xiao, "Traffic sign recognition using kernel extreme learning machines with deep perceptual features," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1647–1653, Jun. 2017.

[43] L. Oneto, E. Fumeo, G. Clerico, R. Canepa, F. Papa, C. Dambra, N. Mazzino, and D. Anguita, "Dynamic delay predictions for large-scale railway networks: Deep and shallow extreme learning machines tuned via thresholdout," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 10, pp. 2754–2767, Oct. 2017.

[44] T. Liu, Y. Yang, G.-B. Huang, Y. K. Yeo, and Z. Lin, "Driver distraction detection using semi-supervised machine learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1108–1120, Apr. 2016.

[45] Y. Jiang, Y. Zhang, C. Lin, D. Wu, and C.-T. Lin, "EEG-based driver drowsiness estimation using an online multi-view and transfer TSK fuzzy system," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1752–1764, Mar. 2021.

[46] O. F. Alcin, F. Ucar, and D. Korkmaz, "Extreme learning machine based robotic arm modeling," in *Proc. 21st Int. Conf. Methods Models Automat. Robot. (MMAR)*, Aug. 2016, pp. 1160–1163.

[47] J. Duan, Y. Ou, J. Hu, Z. Wang, S. Jin, and C. Xu, "Fast and stable learning of dynamical systems based on extreme learning machine," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 6, pp. 1175–1185, Jun. 2019.

[48] Q. She, B. Hu, H. Gan, Y. Fan, T. Nguyen, T. Potter, and Y. Zhang, "Safe semi-supervised extreme learning machine for EEG signal classification," *IEEE Access*, vol. 6, pp. 49399–49407, 2018.

[49] C. Sun, Y. Yu, H. Liu, and J. Gu, "Robotic grasp detection using extreme learning machine," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Zhuhai, China, Dec. 2015, pp. 1115–1120.

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

[51] E. Gelenbe, "Random neural networks with negative and positive signals and product form solution," *Neural Comput.*, vol. 1, no. 4, pp. 502–510, Dec. 1989.

[52] A. U. Bhat, S. S. Merchant, and S. S. Bhagwat, "Prediction of melting points of organic compounds using extreme learning machines," *Ind. Eng. Chem. Res.*, vol. 47, no. 3, pp. 920–925, Feb. 2008.

[53] Y. Wang, F. Cao, and Y. Yuan, "A study on effectiveness of extreme learning machine," *Neurocomputing*, vol. 74, no. 16, pp. 2483–2490, Sep. 2011.

[54] T. Minemoto, T. Isokawa, H. Nishimura, and N. Matsui, "Feed forward neural network with random quaternionic neurons," *Signal Process.*, vol. 136, pp. 59–68, Jul. 2017.

[55] D. S. Broomhead and L. David, "Radial basis functions, multi-variable functional interpolation and adaptive networks," Roy. Signals Radar Establishment, Malvern, U.K., Tech. Rep. ADA196234, Mar. 1998.

[56] Y. Jiang, J. Xue, R. Wang, K. Xia, X. Gu, J. Zhu, L. Liu, and P. Qian, "Seizure recognition using a novel multitask radial basis function neural network," *J. Med. Imag. Health Informat.*, vol. 9, no. 9, pp. 1865–1870, Dec. 2019.

[57] K. Tyagi, C. Rane, B. Irie, and M. Manry, "Multistage Newton's approach for training radial basis function neural networks," *Social Netw. Comput. Sci.*, vol. 2, no. 5, p. 366, Sep. 2021, doi: 10.1007/s42979-021-00757-8.

[58] M. B. Li, G. B. Huang, P. Saratchandran, and N. Sundararajan, "Channel equalization using complex extreme learning machine with RBF kernels," in *Advances in Neural Networks-ISNN*, J. Wang, Ed. Berlin, Germany: Springer-Verlag, 2006, pp. 114–119.

AKIRA SASOU (Member, IEEE) received the B.E., M.E., and Ph.D. degrees in electrical engineering from Tokyo Denki University, in 1994, 1996, and 1999, respectively. He currently leads the Signal Processing Research Group, Department of Information Technology and Human Factor, National Institute of Advanced Industrial science and Technology (AIST).