

# Dynamic Recognition of Environmental Sounds with Recurrent Neural Network

\*Yang Zhang, Tetsuya Ogata, Toru Takahashi, and Hiroshi G. Okuno (Kyoto Univ.)

**Abstract**— This paper introduces our method for classifying non-speech environmental sounds. Most existing studies require a huge number of sample data of all target sounds including noises for training stochastic models such as gaussian mixture model. We propose a use of neuro-dynamical model which can be trained with a small amount of data. In this paper, we show the results of preliminary experiments of the proposed model, which enables classification of both known and unknown sound targets.

**Key Words:** MTRNN, Prediction, Classification, MFCC

## 1. Introduction

Recently, there have been a growing number of studies focusing on systems for classification of environmental sounds. For example, Asakawa developed the model recognizing the sound of writing with chalk to detect writing movement on a board [1]. Ishihara et al. developed the system converting environmental sounds into onomatopoeia [2]. Environmental sound contains a large amount of information, such as tell what happened around here. Therefore it could be a powerful sensor modality for autonomous system working in a real world.

The purpose of this study is to develop a system that enables robots to understand environmental sounds.

Some existing studies for classifying environmental sounds aim to remove the noises. Nakamura developed the information guidance system to identify environmental noises and unnecessary utterance [3], and Miki used a hidden markov model (HMM) to discriminate environmental sounds [4].

However, applying the methods to robot systems, there are two fatal problems as follows:

1. The model requires all types of sounds for classification including noises. However, it is almost impossible for robot systems to obtain training data includes all possible sound samples in advance. Therefore it should equip the ability to classify unknown sound classes.
2. The model a large number of learning samples. However, it cannot obtain a large number of learning sound samples due to its hardware durability.

To solve these problems, we propose the use of dynamical system for classifying environmental sounds. More specifically, in this paper, we used the multiple timescale recurrent neural network (MTRNN) model as a classifier. This model is introduced as the predictor of environmental sounds, and classify not only trained sound but also untrained sounds by its generalization and self-organizing capability.

## 2. MTRNN Model

This section explains the detail of the architecture and learning process of MTRNN model.

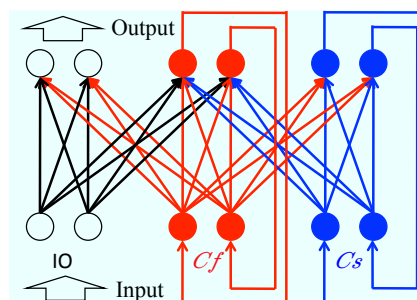


Fig.1 Multiple Timescale Neural Network

### 2.1 MTRNN

The structure of the MTRNN is an extension of a Continuous time recurrent neural network (NN), which combines the neuron groups of which time scales are different [5].

In multiple timescale RNN, recurrent nodes have different changing rate which are controlled by time scale coefficients. Figure 1 illustrates the detailed structure. More specifically, the nodes have a high changing rate (fast context) which can help to generate dynamics, or a low changing rate (slow context), which can help the self-organizing gate to switch structure of primitive sequence data. Each primitive sequence is encoded into the initial value of slow context. Then, we can also generate novel primitive sequence by using this generalized space.

### 2.2 MTRNN Structure

As illustrated in Figure 1, nodes, combination weights, and time scale are three important elements in an MTRNN. Nodes are formed by input/output nodes ( $IO$ ), fast context nodes ( $Cf$ ), slow context nodes ( $Cs$ ). Combination weights are between nodes, except between  $IO$  and  $Cs$ . The time scales of  $IO$ ,  $Cf$ , and  $Cs$  are set differently to keep different changing rates.

### 2.3 MTRNN Equation

The MTRNN can generate sequence data in the forward calculating step. It also can recognize sequence data by using prediction error back propagated through time steps (called back propagation through time).

**Variable Definition:**

**number of nodes:**  $N$

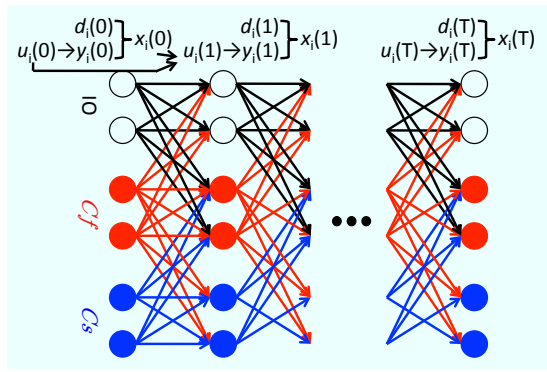


Fig.2 Forward Calculating

**number of IOs:**  $O$

**step number of sequence data:**  $T$

**learning data:**  $d_i(t)$  ( $t = 0, \dots, T$   $i = 1, \dots, O$ )

**time scales:**  $\tau_i$  ( $i = 1, \dots, N$ )

**status:**  $u_i(t)$  ( $t = 0, \dots, T$   $i = 1, \dots, N$ )

**output:**  $y_i(t)$  ( $t = 0, \dots, T$   $i = 1, \dots, N$ )

**input for next step:**  $x_i(t)$  ( $t = 0, \dots, T$   $i = 1, \dots, N$ )

**combination weights:**  $w_{ij}$  ( $i = 1, \dots, N$   $j = 1, \dots, N$ )

### 2.3.1 Forward calculating step

Figure 2 illustrates forward calculating step.  $x_i(t-1)$  and  $u_i(t-1)$  in the  $t-1$  step are determined by  $u_i(t)$  in  $t$  step, output  $y_i(t)$  is determined by  $u_i(t)$ , and input  $x_i(t)$  for  $t+1$  step is determined by  $y_i(t)$  and  $d_i(t)$ .

The forward calculating equations are as follows.

if  $i \in O \wedge j \in Cs$ , or if  $i \in Cs \wedge j \in O$ , then  $w_{ij} = 0$ .

$$u_i(t) = \left(1 - \frac{1}{\tau_i}\right)u_i(t-1) + \frac{1}{\tau_i} \left[ \sum_{j \in N} w_{ij}x_j(t-1) \right] \quad (1)$$

$$y_i(t) = \text{sigmoid}(u_i(t)) \quad (2)$$

$$\text{sigmoid}(x) = \frac{1}{1 + \exp(-x)} \quad (3)$$

$$x_i(t) = \begin{cases} \beta \times y_i(t) + (1 - \beta) \times d_i(t) & i \in O \\ y_i(t) & \text{otherwise} \end{cases} \quad (4)$$

### 2.3.2 Back Propagation Through Time

In BPTT algorithm, input/output values and combination weights are retained while calculating the forward step from 0 through  $T$  steps. After this, it will renew combination weights by back-propagated the prediction error from 0 through  $T$  steps.

We use the sum square error as the error function as follows.

$$E = \frac{1}{2} \sum_{t=1}^T \sum_{i \in O} (y_i(t) - d_i(t))^2 \quad (5)$$

The purpose of BPTT is to minimize error. For this purpose, we calculate partial error differentiation by using combination weights  $w_{ij}$  as follows.

$$\frac{\partial E}{\partial w_{ij}} = \sum_t \frac{1}{\tau_i} \frac{\partial E}{\partial u_j(t)} x_j(t-1) \quad (6)$$

To calculate  $\frac{\partial E}{\partial w_{ij}}$ , we must know  $\frac{\partial E}{\partial u_j(t)} x_j(t-1)$ , which has two cases in which IO nodes and Cf/Cs nodes are calculated as follows.

$$\frac{\partial E}{\partial u_i(t)} = \begin{cases} (y_i(t) - d_i(t))y_i(t)(1 - y_i(t)) + \left(1 - \frac{1}{\tau_i}\right) \frac{\partial E}{\partial u_i(t+1)} & (i \in O) \\ \sum_{j \in N} \frac{\partial E}{\partial u_j(t+1)} \left[ \delta_{ij} \left(1 - \frac{1}{\tau_i}\right) + \frac{1}{\tau_j} w_{ji} y_j(t)(1 - y_j(t)) \right] & (i \in Cf \text{ or } i \in Cs) \end{cases} \quad (7)$$

The following equation is for renewing combination weights.

$$w_{ij}(n+1) = w_{ij}(n) - \alpha \frac{\partial E}{\partial w_{ij}} \quad (8)$$

$\alpha$  is the learning rate constant

## 2.4 Learning, Recognition and Prediction

MTRNN processes sequence data with learning, recognition and prediction phases.

- **Learning:** The MTRNN renews combination weights and the initial values of  $Cs$  until the prediction error (sum square error between learning data and forward calculating) converges by using BPTT. In this phase, sequence data can become self-organized and construct a  $Cs$  space.
- **Recognition:** In the recognition phase, we only use the prediction error to renew the initial values of  $Cs$  (fixed combination weights of BPTT). As a result, we can identify all sequence data as points in  $Cs$  space.
- **Prediction:** In the prediction phase, we can assign an initial values of both sequence data and  $Cs$  and associate all the sequence data with the MTRNN (using forward calculating).

## 3. Environmental Sound Classification System

Figure 3 illustrates the environmental sound classification system. This system can classify known and unknown sounds.

### 3.1 Learning Sounds

Figure 3 illustrates the environmental sound learning flow. There are two different types of learning, “environmental sounds learning using an MTRNN” and “ $Cs$  space classification learning using an NN”.

First, an MTRNN is trained with several environmental sound classes, i.e. it predicts environmental sounds and renews itself based on the prediction error. Simultaneously, environmental sounds will become self-organized at the  $Cs$  space. Learning data is constructed using the mel-frequency cepstrum coefficient (MFCC) (12 dimensions) sequence data from environmental sounds. Next, a three-layer NN is learn the  $Cs$  vector (as input) and class labels (as output). Then we can obtain cluster information of the  $Cs$  space.

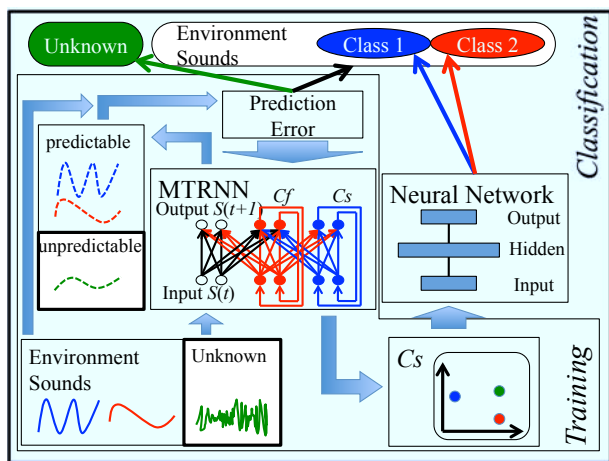


Fig.3 Training and Classification of Proposed System

### 3-2 Classification between target and unknown sound classes

Figure 3 illustrates the recognition flow. We used classification to test, if the sound class had learned before. More specifically, the MTRNN first calculates the prediction error (sum square error between sound and predicted sound), then classifies target and unknown sound classes basing on the threshold value of the prediction error. We expect that the prediction error of unknown sounds is larger than others, since the MTRNN did not learn sounds of same classes before.

### 3-3 Classifying target sound classes

Figure 3 illustrates the recognition flow. We use classification to obtain class labels of the sounds from many environmental sound classes. The MTRNN recognizes the input sounds then identify them with the initial value of  $C_s$ , and then a three-layer NN gets class labels of the  $C_s$  vector from the results as input.

## 4. Experiment

We examined the “Classification between target and unknown sound classes” and the “Classifying target sound classes” using environmental sounds.

### 4.1 Experiment Conditions

We used the MFCC for extracting environmental sounds with a 25 ms window and 10 ms interval. Then the processed MFCC through the smoothing and normalizing steps. Finally, the results construct learning data.

Learning data contained five classes of environmental sounds, Bell, Cyclebell, Glass, Gun, and Whistle. Bell, Glass and Whistle contained 100 samples each. Glass contained 42 samples, and Whistle contained 180 samples.

First we picked ten samples from each class as learning data, and the rest as evaluation data. In this experiment, we examined the results of the “Classification between target and unknown sound classes” using five-fold cross validation. More specifically, each time of examining, there was one unknown class as unknown class, the other four classes were the target sound classes.

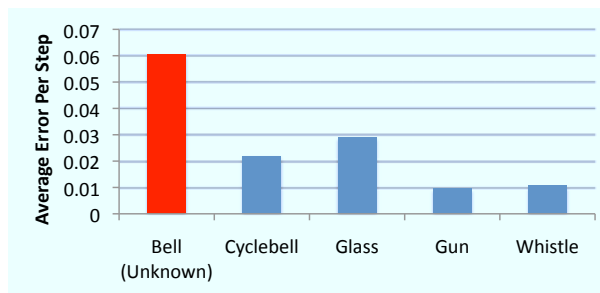


Fig.4 Average Prediction Errors

Table 1 Classification Accuracy of Target and Unknown Environmental Sounds

Unknown	Threshold [%]	Target Env. [%]	Unknown [%]	Average [%]
Bell	0.03	92	99	95
Cyclebell	0.03	99	100	99
Glass	0.02	93	100	96
Gun	0.01	71	100	85
Whistle	0.09	99	100	99

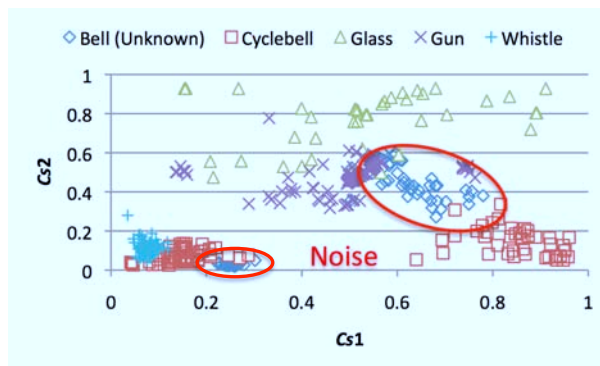


Fig.5 Distribution of First 2 Elements of  $C_s$  Vector

## 4.2 Experimental Result

### 4.2.1 Classification between target and unknown sound classes

The system predicted all sounds of the environmental sound classes, and calculated the prediction error. Figure 4 illustrates the average prediction error when Bell was specified the unknown sound class. The prediction error of Bell was larger than other classes. Therefore we can use this feature to classify target and unknown sound classes.

The results of classification based on a threshold value of the prediction error are listed in table 1. We define average classification accuracy as follows.

**average classification accuracy** =  $0.5 \times (\text{target sound classification accuracy} + \text{unknown sound classification accuracy})$

Table 1 lists results of the highest average classification accuracy.

### 4.2.2 Classification of target sound classes

We trained the NN using the  $C_s$  vector of target environmental sounds. Figure 5 illustrates a distribution map, which shows the first two elements of the  $C_s$  vector (total of five elements). Although the MTRNN did not learn sounds of bell class before, it constructed a cluster of bell classes at the  $C_s$  space. We can see that environmental sounds self-organizes through MTRNN in the  $C_s$  space.

**Table 2** Classification Accuracy of Environmental Sounds

Unknown	Target Env. [%]				
	Bell	Cyclebell	Glass	Gun	Whistle
Bell		90	88	99	84
Cyclebell	100		97	91	100
Glass	100	100		100	100
Gun	96	100	97		99
Whistle	100	98	78	98	

Figure 2 illustrates the results of five-fold cross validation.

#### 4.3 Discussion

- Determining threshold value for classification  
Classification of target sounds and unknown sounds is based on the prediction error. Since performance will be much influenced by the threshold value, designing an effective selection technique to determine threshold value is necessary.
- Classification of  $C_s$  space  
Figure 5 illustrates that target and unknown environmental sound classes are self-organized in the  $C_s$  space. Classifying the  $C_s$  space not only discover target environmental sound classes, but also enables the discovery of unknown sound classes. In this experiment, we used a three-layer NN to classify the  $C_s$  space for convenience. A more effective method for classification is needed in the future.
- Comparison and integration of gaussian mixture model (GMM) technique  
Our method could classify unknown sound classes, but requires a long time for learning calculating compared with the GMM technique. In the future, we will estimate the performance of GMM and integrate with GMM [6].

#### 5. Future work

1. Automatic searching for  $C_s$  space  
The MTRNN can associate environmental sounds from any coordinates of the  $C_s$  space. We are considering a new technique to determine the boundaries of environmental sound classes in the  $C_s$  space. We want to use learning data as the prototype. This is the same problem as clustering analysis for associating data.
2. Comparison with GMM  
We will estimate the performance of GMM to compare our system using a small amount of learning data.
3. Classifying unknown sound classes  
We confirmed the self-organizing state of several unknown sound classes. This shows the possibility for building clusters of unknown sound classes.
4. Learning method  
Sakaguchi adopted neural network approach for classifying environmental sounds [7]. Environmental sounds were converted to 24 channel pulses using the cochlea model, then these pulses were learned using a pulse neural network. The aim of this study was not classifying unknown sounds directly, but show-

ing a way to reduce the reaction of band noise that restrains activity of optimum frequency neurons from other neurons. We will examine the possibility to import the same mechanism to the MFCC.

#### 6. Conclusion

We constructed a environmental sounds classification system using dynamical model and estimated performance using five classes environmental sounds. In experiment, we confirmed the self-organizing of environmental sound classes using little learning data, and the classification of target environmental sound classes and unknown sound classes.

In the future, we will implement a technique of searching the  $C_s$  space with a small amount of learning data, examine the classification of unknown sound classes.

**Acknowledgment** This research supported by JST PRESTO (Information Environment and Humans), Grant-in-Aid for Creative Scientific Research (19GS0208), and Grant-in-Aid for Scientific Research (B) (21300076).

#### References

- [1] Taira Ashikawa, Akira Sugauma, Rin-ichiro Taniguchi: "Development and Validation of an Automatic Camera Control System Made with a Detection of a Chalking Sound", IEICE. ET, Vol. 102, No. 509, pp. 43-48(2002).
- [2] Kazushi Ishihara, Tomohiro Nakatani, Kazunori Komatani, Tetsuya Ogata, Hiroshi G. Okuno: "Designing Environmental Phoneme for Automatic Sound-Imitation Word Recognition", Proceedings of the Annual Conference on JSAI (CD-ROM), Vol. 18, pp. 1E2-03(2004).
- [3] Keisuke Nakamura, Ryuichi Nisimura, Akinobu Lee, Hiroshi Saruwatari, Kiyohiro Shikano: "Identification of environmental noise and unnecessary utterance on a real information guidance system with spoken dialogue interface", IEICE. SP, Vol. 103, No. 632, pp. 13-18(2004).
- [4] Kazuhiro Miki, Takanobu Nishiura, Kiyohiro Shikano: "Environmental Sound Discrimination Based on Hidden Markov Model", IEICE. SP, Vol. 99, No. 525, pp. 79-84(1999).
- [5] Y. Yamashita, J. Tani: "Emergence of functional hierarchy in a multiple timescale neural network model: A humanoid robot experiment", PLoS Computational Biology, Vol. 4, No. 11(2008).
- [6] Nobuyuki Miyake, Tetsuya Takiguchi, Yasuo Arikawa: "Sudden Noise Reduction Based on GMM with Noise Power Estimation", IEICE. SP, Vol. 107, No. 406, pp. 25-30(2007).
- [7] Shinya Sakaguchi, Susumu Kuroyanagi, Akira Iwata: "Sound Discrimination System for Environment Acquisition", IEICE. NC, Vol. 99, No. 494, pp. 61-68(1999),