Hindawi
Advances in Multimedia
Volume 2018, Article ID 3748141, 13 pages
https://doi.org/10.1155/2018/3748141

Research Article

Research on Objective Evaluation of Recording Audio Restoration Based on Deep Learning Network

Cong Jin,[1] **Wei Zhao**,[2] **and Hongliang Wang**[3]

[1]*Key Laboratory of Media Audio & Video, Communication University of China, Beijing, China*
[2]*College of Science and Technology, Communication University of China, Beijing, China*
[3]*Advertising School, Communication University of China, Beijing, China*

Correspondence should be addressed to Cong Jin; jincong0623@cuc.edu.cn

Received 13 March 2018; Accepted 13 June 2018; Published 18 September 2018

Academic Editor: Zechao Li

There are serious distortion problems in the history audio and video data. In view of the characteristics of audio data repair, the intelligent technology of audio evaluation is explored. As the traditional audio subjective evaluation method requires a large number of personal to audition and evaluation, the tester's subjective sense of hearing deviation and sample space data limited the impact of the accuracy of the experiment. Based on the deep learning network, this paper designs an objective quality evaluation system for historical audio and video data and evaluates the performance of the system and the audio signal quality from the perspective of feature extraction and network parameter selection. Experiments show that the system has good performance in this experiment; the predictive results and subjective evaluation of the correlation and dispersion indicators are good, up to 0.91 and 0.19.

1. Introduction

In 1999, Gers et al. introduced the forget gate on the basis of the original LSTM, and after adding the forgotten door, the model had the ability to clear useless historical information.

In 2000, Gers & Schimidhuber et al. [1] believe that if you want to make more accurate use of historical information and input data, the memory cell itself should also participate in the control of multiple doors. Therefore, they made Peephole Connections, the memory cell status as the input of the various doors.

In 2005, Graves & Schmidhuber et al. proposed the training of the LSTM model by using a complete backward propagation approach. The implementation of the LSTM process can be made more reliable because of the addition of gradient detection steps. In 2009, Bayer proposed different LSTM models in order to improve the stability of the model when dealing with some context-sensitive sequence problems.

In 2014, sak et al. in order to reduce the number of parameters of the LSTM model of the multi-memory module, a linear dimensionality reduction operation is added after the LSTM output. And Doetsch et al. improved the performance of LSTM in offline handwriting recognition data sets by introducing new parameters into the activation functions of the various gates. Cho et al. [2] proposed a simplified version of LSTM, GRU (Gated Recurrent Unit). The GRU unit has only two gates, reset gate and update gate. However, after several sets of comparative experiments, it was found that although GRU had fewer parameters, the performance on multiple tasks was similar to that of LSTM.

2. Theory of LSTM

2.1. The Basic Thought and Workflow of LSTM

2.1.1. Forgetting Information. The role of the forgetting gate is to control the function of some unwanted information accumulated in the neural network before it is forgotten. Concretely, its input is a network input Xt at a certain time and one of the outputs of the previous network element, ht-1, whose output is a value between 0 and 1, which is used to communicate with the previous network element. The state Ct-1 is multiplied, as shown in Figure 1.

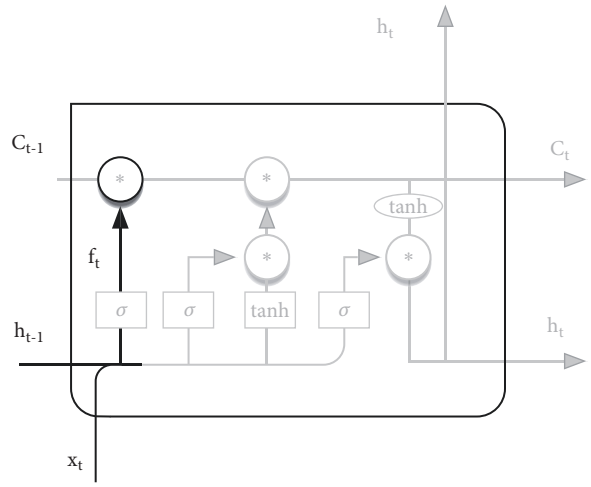

Figure 1: Forgetting gate layer.

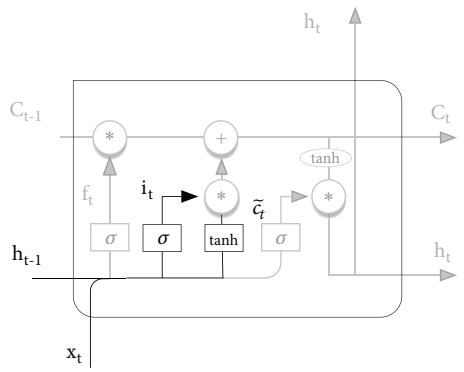

Figure 2: LSTM status update layer.

The process can be represented by the following formula:

$$f_t = \sigma\left(W_f\right).\left[h_{t-1}, x_t\right] + b_f \tag{1}$$

2.1.2. Update Status. Once the network controls the degree of forgetting for the previous "memory", the next step is to control the amount of information stored in each network element. This step contains two parts: first, use an input gate (the sigmoid layer in the figure) to control the information update; next, a tanh layer is used to create a "candidate" vector $\widetilde{C}$, which is used to add to the state of the current network unit and become part of it, as shown in Figure 2.

The process above can be represented by the following formula:

$$\begin{aligned} i_t &= \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right) \\ \widetilde{C}_t &= \tan h\left(W_c \cdot \left[h_{t-1}, x_t\right] + b_c\right) \end{aligned} \tag{2}$$

2.1.3. Decision Output. In LSTM, each network cell first performs some 'filtering' operations on the current cell state and then outputs the results as an output. Concretely, the current unit state output section is controlled by a sigmoid layer, and then the result of the previous stage is processed by the tanh layer and multiplied by the output of the sigmoid layer; thereby we can obtain one of the output hidden layers of the network unit, as shown in Figure 3.

The process above can be represented by the following formula:

$$\begin{aligned} O_t &= \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \\ h_t &= o_t * \tanh\left(C_t\right) \end{aligned} \tag{3}$$

2.2. The Variant Structure of LSTM. The structure used by Gers and Schmidhuber in research [3] is shown in Figure 4. As you can see from the figure, this structure adds a structure called "peephole" to the original infrastructure, so that each gate structure can be connected to the current cell state.

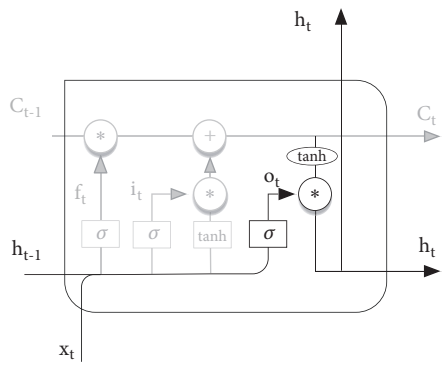

Figure 3: LSTM decision output layer.

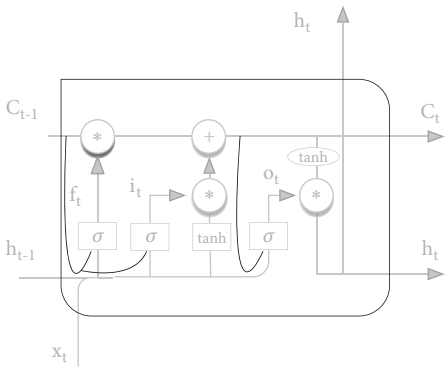

Figure 4: Variant structure 1 of LSTM.

The structure of the workflow is expressed as follows:

$$\begin{aligned} \mathrm{f}_t &= \sigma\left(W_f \cdot \left[C_{t-1}, h_{t-1}, x_t\right] + b_f\right) \\ \mathrm{f}_i &= \sigma\left(W_f \cdot \left[C_{t-1}, h_{t-1}, x_t\right] + b_i\right) \end{aligned} \tag{4}$$

Another structure considers the structure of a pair of mutually associated forgetting gates and output gates. Unlike all previous structures, the structure no longer considers the state of control and update of the forgotten information separately, but only the former, which shows that the structure only outputs the unit state values after forgetting the useless information. Its structure is shown in Figure 5.

The process above can be represented by the following formula:

$$C_{\mathrm{t}} = f_t * C_{t-1} + \left(1 - f_t\right) * \widetilde{C}_{\mathrm{t}} \tag{5}$$

In study [4], Cho proposed a more dynamic LSTM variant structure, GRU, which combines the forgetting gate and the output gate into a single update gate and also fuses the hidden layer and the state in the network cell. Cho has applied the RNN network structure in his research on Machine Translation, which is called RNN codec model. The encoder is responsible for coding a symbol sequence into a fixed length vector representation, and the decoder is responsible for converting the formula to another symbol sequence. He uses this codec model as an additional feature and combines traditional logarithmic linear models to achieve better translation results. The structure is shown in Figure 6.

The operation is expressed as follows:

$$\begin{aligned} Z_{\mathrm{t}} &= \sigma\left(W_z \cdot \left[h_{t-1}, x_t\right]\right) \\ r_t &= \sigma\left(W_r \cdot \left[h_{t-1}, x_t\right]\right) \\ \widetilde{h}_{\mathrm{t}} &= \tanh\left(W \cdot \left[r_t * h_{t-1}, x_t\right]\right) \\ h_t &= \left(1 - z_t\right) * h_{t-1} + z_t * \widetilde{h}_t \end{aligned} \tag{6}$$

3. System Experiment and Analysis

3.1. System Composition Framework. As shown in Figure 7, the whole framework of the speech quality evaluation system is presented, which can be divided into four stages: preprocessing stage, feature extraction stage, training stage, and system evaluation phase.

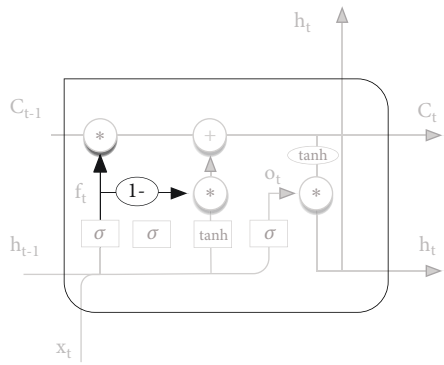

Figure 5: Variant structure 2 of LSTM.

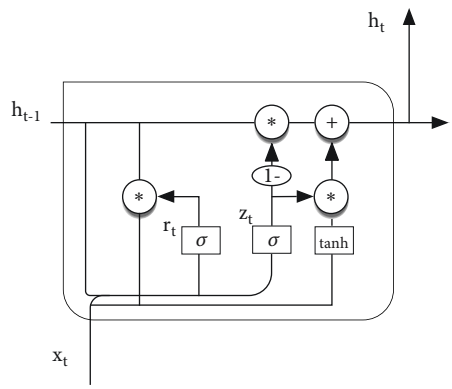

Figure 6: Variant structure 3 of LSTM.

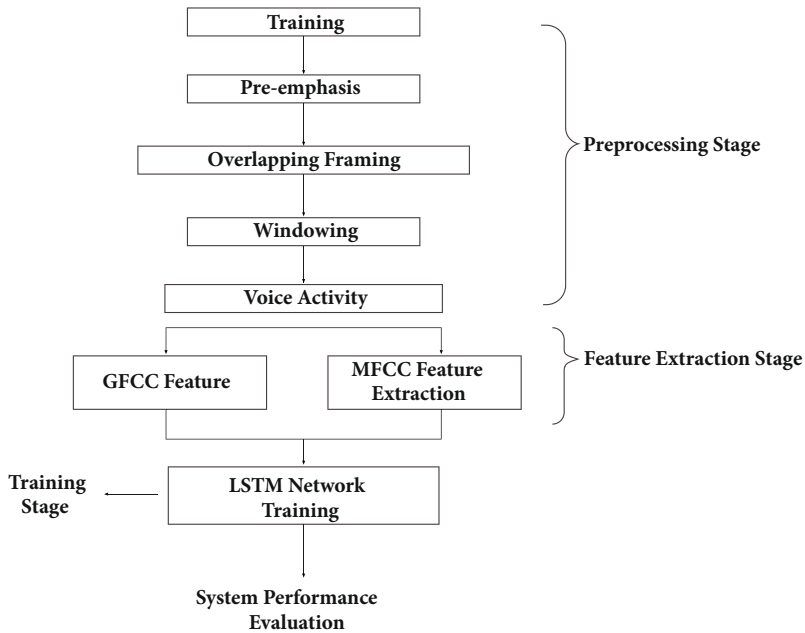

Figure 7: System composition framework.

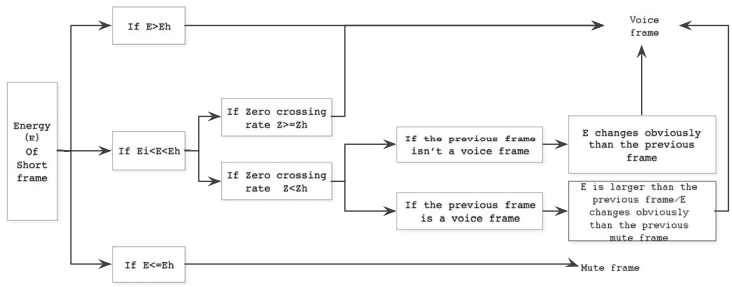

Figure 8: Voice activity detection process.

It can be seen from the figure, the system first preprocesses speech, including preemphasis, overlapping frames, windowing, and the steps of voice activity detection, then the processed data are extracted by artificial feature extraction, the system using GFCC. And the MFCC characteristics were compared with the reference control group. The feature data are trained in the LSTM neural network, and the classification of the samples is used as the label of classification. After training the network model, the quality of the speech samples in the sample space is evaluated (the classification results are given) and compared with the sample real tags, so as to evaluate the performance of the system.

Each module of the system is described in detail below.

3.1.1. Preprocessing Module. The preprocessing module mainly includes speech signal preemphasis, windowing and subframe and voice activity detection. Among them, the signal preemphasis, by canceling the extra poles of glottal excitation, strengthens the high frequency part of the speech signal, thus smoothing the speech signal and providing convenience for analyzing the spectrum and sound channel parameters; windowing is to divide the whole speech signal into several short segments, thus satisfying the assumption that the short-time speech signal is stable, in which the length of each frame is 25ms, and the overlap time is 10ms; voice activity detection is to use short-time energy combined with short-time zero rate as the standard to evaluate whether a frame signal is speech frame, by removing the silence segment, the response of data to speech signal feature is more significant, and improve the accuracy and reliability of the follow-up process for speech quality evaluation. The specific process of voice activity detection is shown in Figure 8.

Among them, Eh, El, Zh, and Zl, respectively, indicate the frame short-time energy and zero crossing rate threshold.

3.1.2. Feature Extraction Module. The speech feature selected in this paper is the traditional MFCC feature and the improved GFCC feature. The extraction process of the two features is described in detail below and compared.

Compared with the GFCC feature, the MFCC uses the triangular filter banks with n center frequency distribution in the Mel frequency domain to filter the spectrum energy, which is called the Mel filter bank. The operation of this process mainly plays a role of smooth spectrum, eliminates the harmonics, and highlights the formant in the original speech and through this approach highlights the core information in the original signal and eliminates the influence of different tone interference information; another difference is the use of logarithmic compression when compression is performed.

The comparison of the two compression functions is shown in Figure 9.

3.1.3. Network Training Module. In this experiment, after the training voice fragments are feature extracted, we sent them into the neural network with LSTM as the structure. And the number of network units circle is 400, the characteristic matrix through the input of hidden layer integration and then entering the LSTM neural units to train; on the basis of the last layer: LSTM neural unit, the probability distribution is obtained by Softmax, and the maximum probability of dimension is selected as the final classification forecast.

3.1.4. System Evaluation Module. The evaluation system can be divided into two categories: for the evaluation of neural network structure and the training results, we mainly start with the network parameters, and, through experiments on different parameters, we can select the best value group of parameters for the network; for the evaluation of feature selection method, we mainly start with the feature extraction, through the training of different character data, to compare the different effect of different characteristics in the quality evaluation.

Among them, the former considered parameters including the learning rate, dropout and network layers, and the index of evaluation is the final correct rate for the network on the test set; the latter mainly compare GFCC and MFCC, and the index of evaluation is correlation coefficient (Karl-Pearson) and root-mean-square error (RMSE). These two parameters are used to describe the degree of proximity and fluctuation between the objective voice quality of the system estimate and the given subjective mass fraction. The following is to briefly introduce these indexes.

(a) Correct Rate. We separate some sample data into a test sample set, which does not participate in network training,

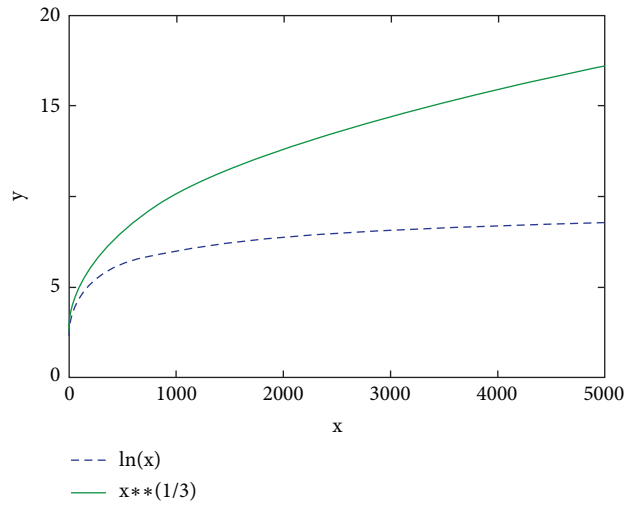

Figure 9: Cube root compression and logarithmic compression.

and just is used to finally test the generalization and robustness of the network model. Specifically, we compared each sample quality score of the network prediction with the actual given quality score, and calculated the proportion of the same number of samples in the whole test set, and then obtained the accuracy of the test set. The higher the correct rate is, the more the network parameter setting meets the training task.

(b) Karl-Pearson Coefficient. The expression is as follows:

$$R = \frac{\sum_{i=1}^{N} (\lambda_i - \mu_\lambda)\left(\widetilde{\lambda}_i - \mu_\lambda\right)}{\sqrt{\sum_{i=1}^{N} (\lambda_i - \mu_\lambda) \sum_{i=1}^{N} \left(\widetilde{\lambda} - {\mu_\lambda}^2\right)}} \tag{7}$$

λ_i represents the subjective quality evaluation score given by the human ears, and the $\widetilde{\lambda}_i$ represents the predictive quality evaluation scores given by the system, λ_i, $\widetilde{\lambda}_i$ represent the average values of λ_i, $\widetilde{\lambda}_i$ respectively. N represents the number of samples. The range of R is $[0, 1]$ which is to measure the correlation between the predicted quality scores and the actual subjective scores.

(c) RMSE. The mean square error RMSE is mainly used to measure the estimation accuracy of the system. The expression is as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \left(\lambda_i - \widetilde{\lambda}_i\right)^2}{N}} \tag{8}$$

The meanings of each variable are the same as those in formula (8). In practice, 0.5 is regarded as distinction threshold in the most: when the RMSE is smaller than the value, the system accuracy is higher, the subjective evaluation is smaller, otherwise, the accuracy is lower.

3.2. Voice Library and Experimental Environment

3.2.1. Voice Library. We obtained 300 pieces of old film voice from the film museum, which is including speech, music, voice singing, car sound, war artillery, musical instruments, six categories, each category contains 50 segments. Each segment of audio is 10s long, and the film archive's restoration personnel provided us both the same soundtrack of the audio before and after (the restoration), and the quality evaluation scores.

3.2.2. Experimental Environment. The experimental environment is built with Tensorflow neural network framework. Tensorflow is an open source numerical calculation software library based on data flow diagrams. The nodes in the flow diagrams represent mathematical operations, while the edges represent the multidimensional data array of that is connected in the operation (it is called tensors in Tensorflow). Tensorflow, originally developed by Brain Team of Google machine intelligence research organization, which is for machine learning and deep neural network research and development and has been widely used and popularized now. In this experiment, we selected Tensorflow 1.01 version for the study.

3.2.3. Parameter Adjustment

(a) Learning Rate Setting. In the network training process, the weights are updated according to the following formula:

$$weights = weights\text{–}step\ size * weights\ grad \tag{9}$$

The learning rate plays a role in determining the weight and the rate of updating. If the setting is too large, it is easy to cause the weights to exceed the optimal value and swing around the optimal value. If the setting is too small, it will cause slower network training. Because the problem that the down of standard gradient is easy to fall into the local optimal solution.

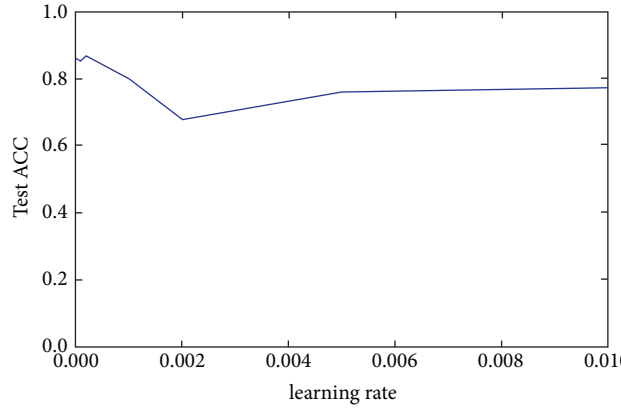

Figure 10: Correct rate of test set under different learning rates.

Table 1: Correct rates of test set under different learning rates.

Learning rate	2e-6	5e-6	1e-5	2e-5	5e-5	1e-4
Correct rate	0.602	0.694	0.741	0.829	0.853	0.849
Learning rate	2e-4	5e-4	1e-3	2e-3	5e-3	1e-2
Correct rate	0.842	0.830	0.790	0.806	0.695	0.732

(b) Dropout. Dropout optimization is a way to improve network generalization ability and reduce overfitting. By setting the dropout parameters, it makes the hidden nodes of network training process output to a probability value of 0, at the same time, when using the back-propagation algorithm to update the weights, which are connected with the node weights do not participate in the update operation. A figurative metaphor is: By "kill" hidden nodes in the network randomly, the remaining nodes will become more efficient and more reliable.

Specifically, for multilayer LSTM network structures, dropout acts only between layers and layers at the same time, and does not apply dropout to steps at different times between the same layers. In this experiment, we first set dropout parameters which are more central, and then gradually expand to the extreme value. According to different results, we determine the most appropriate dropout parameter value.

(c) LSTM's Layers. Using multilayer LSTM can improve the learning effect of the network for the sample features, but it also brings negative effects of increasing computation and running time, and it cannot provide significant training results of the network. In this experiment, the initial selection of network layers is three, and experiments are carried out in the range of 1-7. It will select the optimal parameter values according to the training results.

4. The Results of Experiment Evaluation

4.1. Network Parameter. The method of controlling variables was used in this experiment. First, we refer to some studies of the deep learning to set network parameters; and then by fixed all remaining parameters and for a parameter in a certain range, repeat training experiment; after the optimal parameter value, we compare the next parameters, and maintain optimal parameter settings for before value. After comparing the experimental results, an attempt is made to find the optimal parameter set suitable for the network. The data analysis process is as follows.

4.1.1. Learning Rate. The selected range of learning rate is [10-2,2∗10-6]. We regard the correct rate of the network on the testing set as the ultimate measure of network training quality basis. At the same time, we combined with changes in the process of training on the training set accuracy rate to analyze them. Table 1 and Figure 10 show the relationship between the change of learning rate and the accuracy of the test set in terms of graphs and curves.

After comparison, it can be found that, with the learning rate gradually changing from large to small, the correct rate of test set experience gradually increases and then decreases, and the optimal learning rate is between 2∗10-4 and 2∗10-5.

Figures 11 and 12 compare and analyze the training set under different learning rate; and test the changes of the correct rate on the set in the whole training process and make a brief analysis and explanation.

The correct rate of change, compared with the whole training process can be found, when the learning rate is far greater than the optimal value, the correct rate of the training set and validation set fluctuated greatly, which shows that the algorithm to search the optimal solution over the optimal

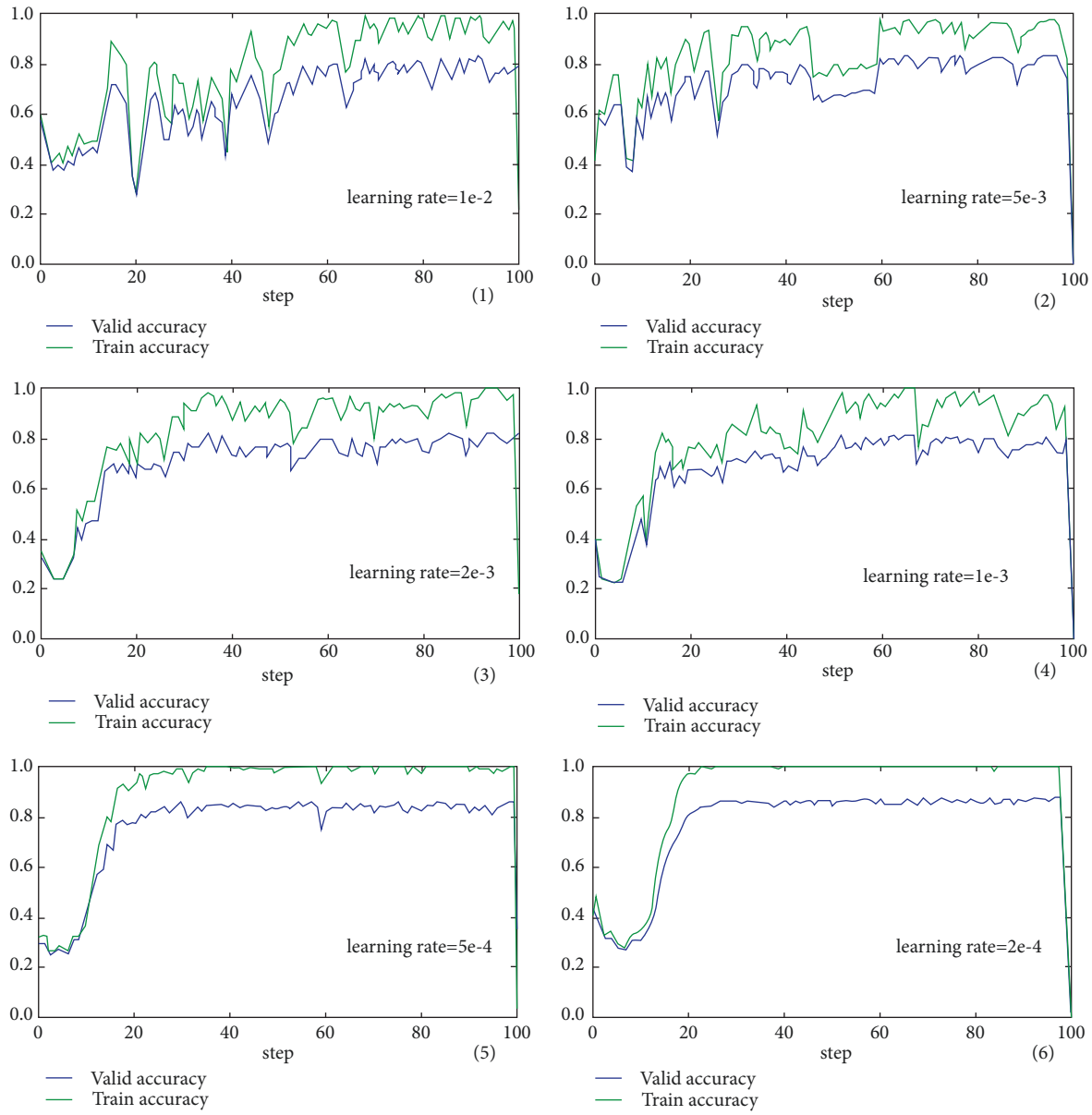

Figure 11: Changes of correct rate under different learning rates in training process (1).

value easily, and result in the process of solving fluctuation near the optimal value; when the learning rate is far less than optimal, training set and validation set correct rate was low, indicating that the learning rate is too small, the algorithm to the optimal value of the convergence process is too slow, resulting in training is not enough to make the algorithm to find the optimal solution; when the learning rate is near optimal value, the correct rate of the training set and validation set is higher, and the convergence is faster, visible in Figure 11 and Figure 12.

Figures 13 and 14 compare and analyze the change of loss function value under different learning rates in the whole training process and make a brief analysis and explanation.

By comparing the changes in the value of the loss function in the whole training process, the following can be found:

When the learning rate is much greater than the optimal value, the loss function value of the training set and the verification set fluctuates greatly, and the downward trend is not obvious. It shows that the algorithm is too modified and produces large oscillation when searching for the optimal solution.

When the learning rate is close to but still greater than the optimal value, the loss function value of the training set and the verification set has a small range fluctuation, and the training begins to decline rapidly, and a slight improvement is made at the end of training. This may be due to the excessive number of training in this kind of learning rate.

When the learning rate is near the optimal value, the change curve of loss function value is smoother, decreases

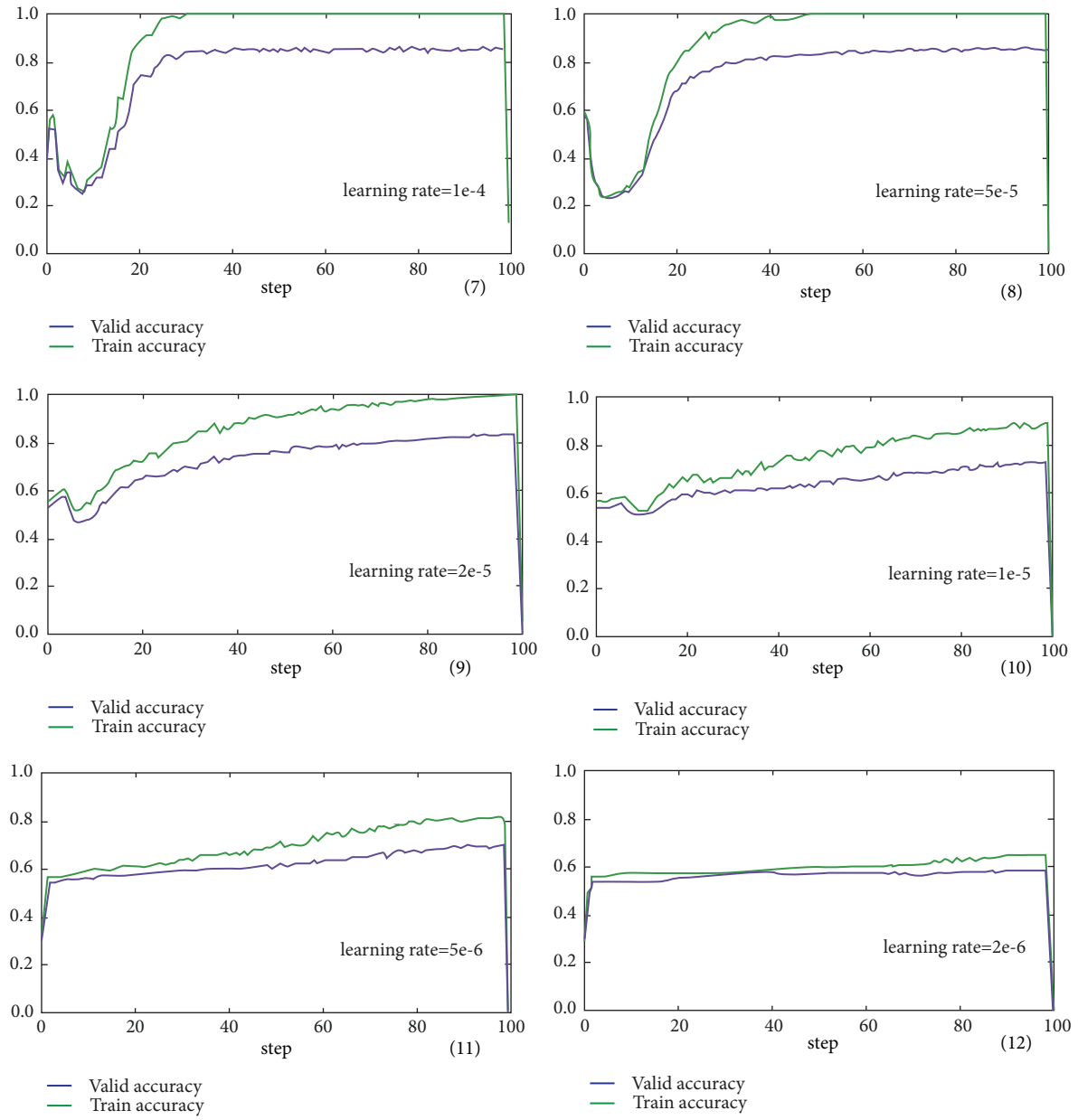

Figure 12: Changes of correct rate under different learning rates in training process (2).

rapidly, and the final fall point value is lower. It shows that the training process converges well.

When the learning rate is less than the optimal value, the value of the loss function declines slowly, and the final point of fall is larger. This shows that the learning rate is too small, and the convergence process of the algorithm to the optimal value is too slow, so that training can not make the algorithm to find the optimal solution.

In combination with the analysis above, the optimal learning rate is finally selected as 5∗10-5.

4.1.2. Dropout Parameters. In the case that the network layer is 4 and the learning rate is 5∗10-5, the dropout parameters are similar, and the results are shown in Table 2.

The training set, and the validation set in training process are shown in Figures 15 and 16.

As it can be seen from the figure, with the increase of dropout parameters, the number of neurons wase "death" in the network gradually reduced, and the data learning ability is gradually strengthened, which is reflected in the correct rate of the training set and validation set faster to reach the optimal value; combined with the set of test results in the table, when the parameter value is 1 (that is, there is no give-up neuron in network), network is slightly over fitting, at the same time when the parameter value is 0.3/0.4 (that is, there are more than half of the probability of giving up neurons), learning ability of network for data feature is slightly weakened. In general, when the values are chosen in

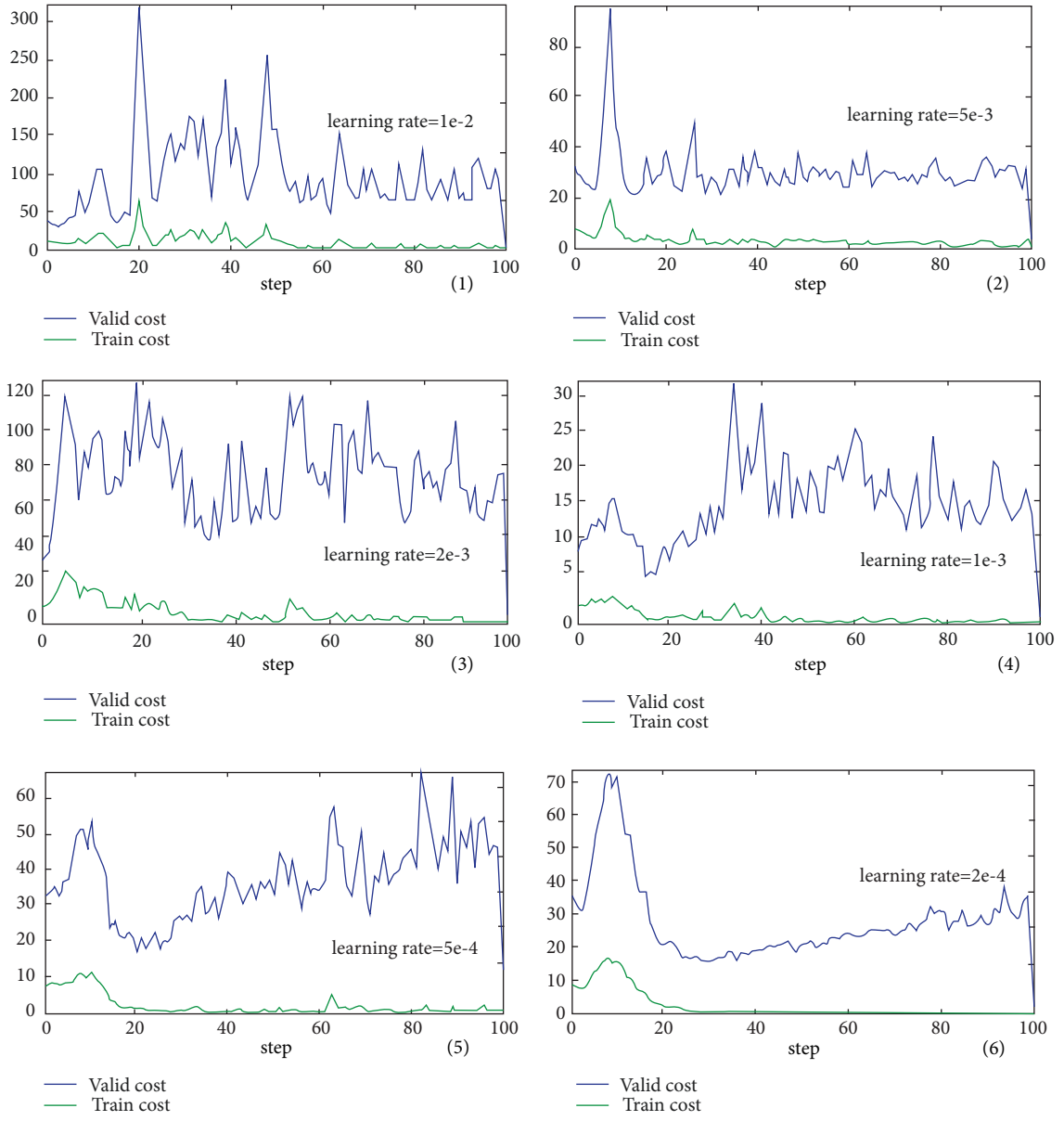

Figure 13: Changes of loss function under different learning rates in training process (1).

Table 2: Correct rate of test set under different dropout.

Dropout	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
correct rate	0.828	0.834	0.855	0.863	0.847	0.853	0.802	0.799

the range of 0.6-0.8 value, it can maximize the optimization of network performance and avoid overfitting. The optimal value of the parameter is finally selected 0.7.

4.2. Feature Selection. The MFCC and GFCC parameters extracted from different orders are trained as features, and the results are shown in Table 3 (the highest values of each result).

By comparing the data in the table, we can summarize as follows: the correlation coefficient of performance, when MFCC features are added in intermediate frequency (6-7 order), the system performance has improved significantly, the high frequency part of the correlation coefficient showed little effect. From the Minimum mean square deviation to see, with the increase of the order of the characteristic coefficients of MFCC, the minimum mean square variance decreases gradually. When joining the high frequency part (order 8-13), the promotion is most obvious. Comparing the different features, we can find that both the correlation and the least mean square variance show that the GFCC feature is superior

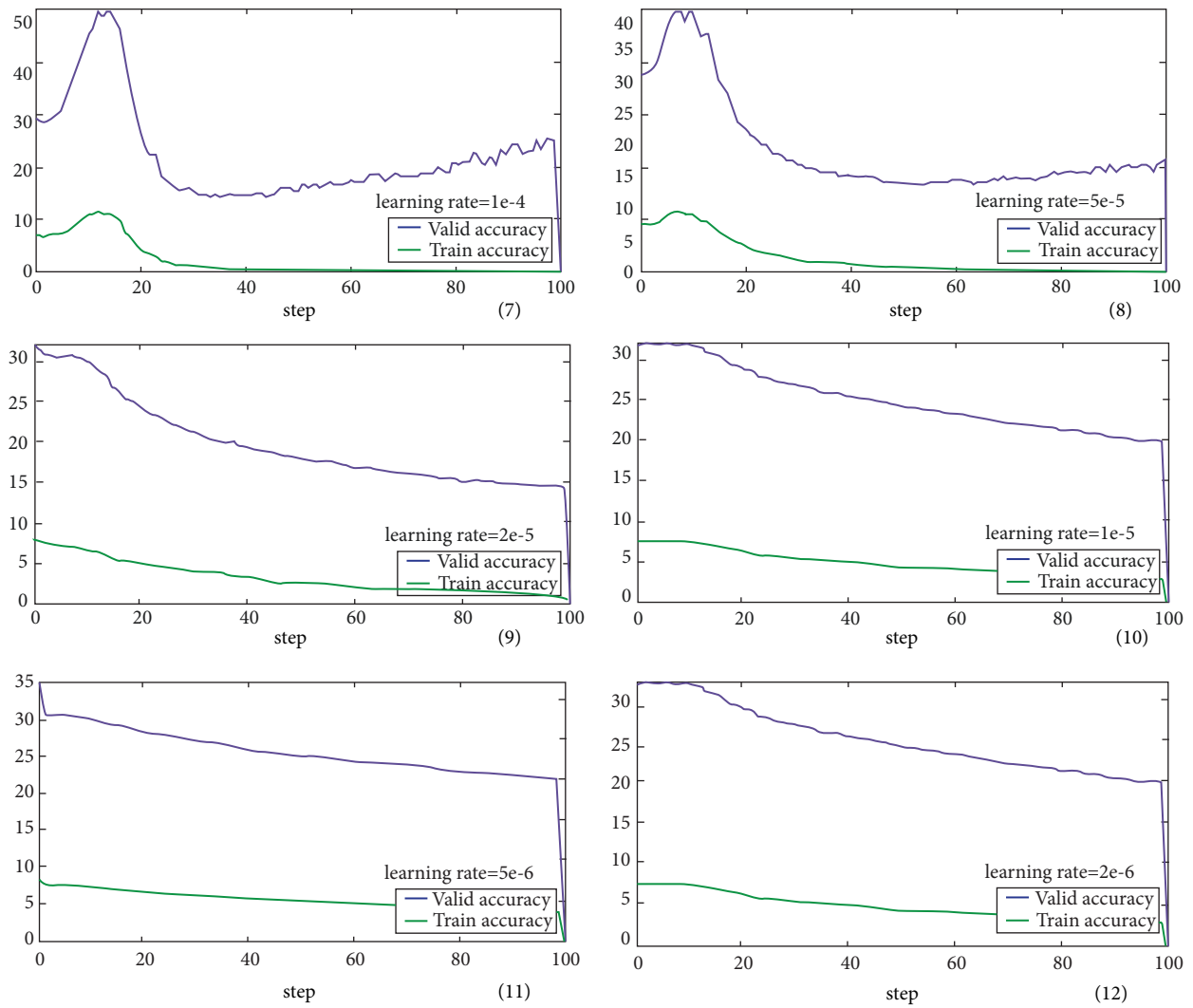

Figure 14: Changes of loss function under different learning rates in training process (2).

Table 3: The results of each index under different audio characteristics.

Feature type	correlation coefficient R	Mean square error RMSE
MFCC (1-2)	0.4369	0.6882
MFCC (1-5dimensions)	0.6929	0.5847
MFCC (1-7dimensions)	0.8015	0.4758
MFCC (1-13dimensions)	0.8343	0.2799
MFCC (1-20dimensions)	0.8909	0.2307
GFCC (13dimensions)	0.9117	0.1924

to the MFCC feature. This shows the former compress by using the Gammtone filter and the cube root, which are very good improvement for the latter and better simulate the physiological characteristics of the human ear.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

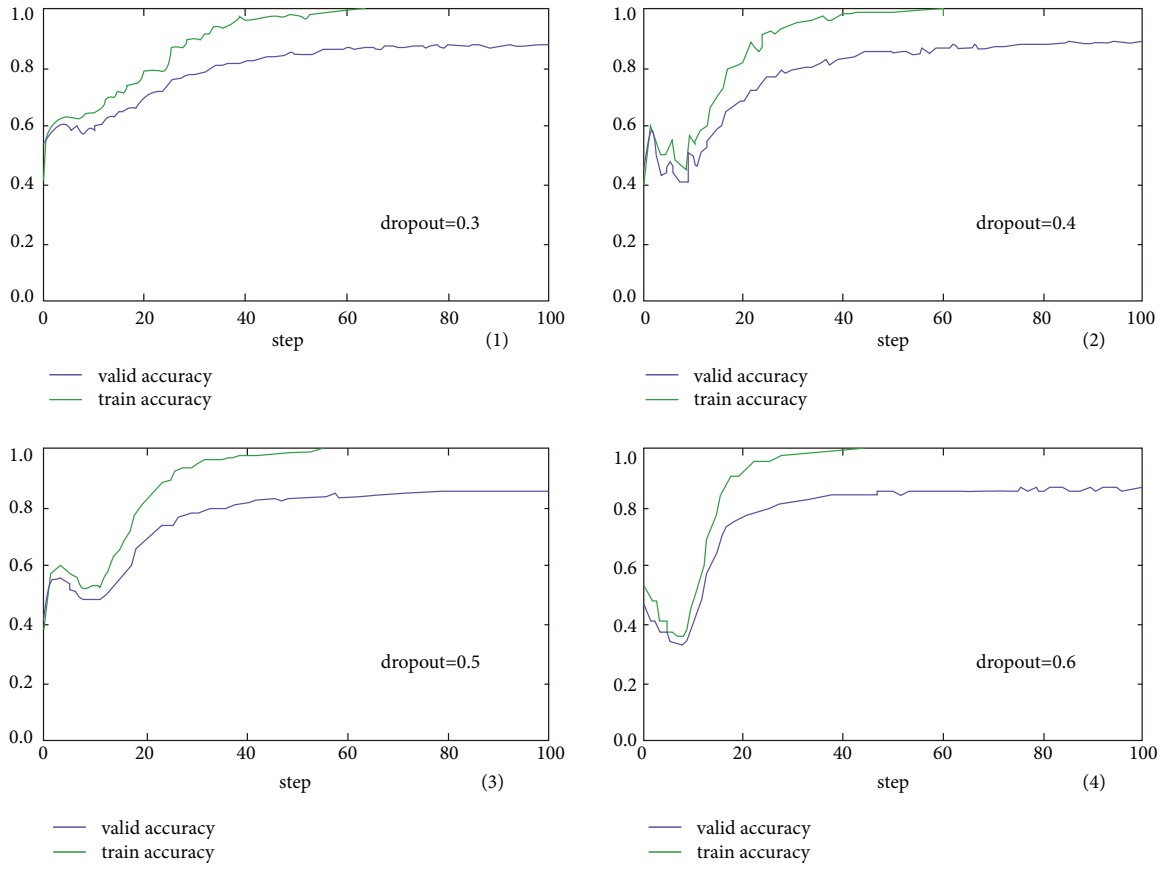

Figure 15: Changes of correct rate under different dropout in training process (1).

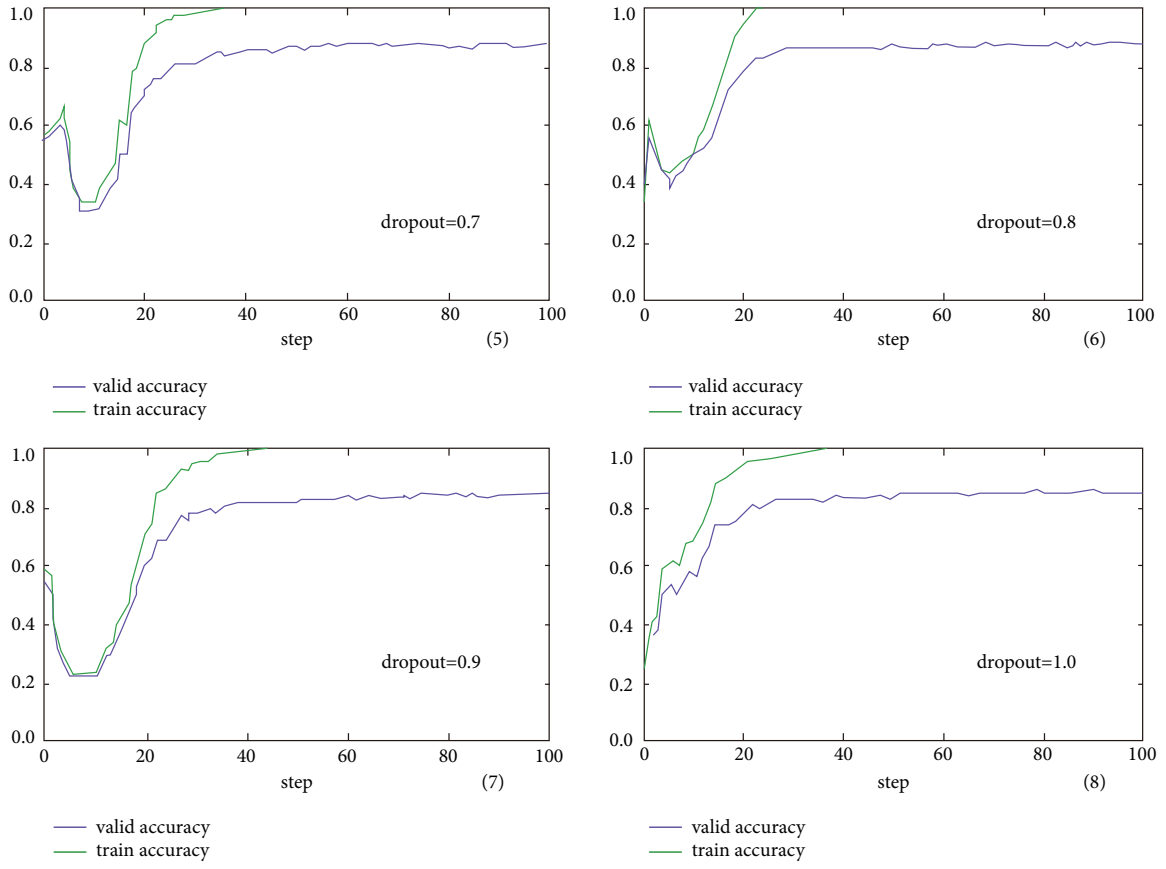

Figure 16: Changes of correct rate under different dropout in training process (2).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The paper is sponsored by NSFC Key Funding no. 61631016; sponsored by High-Grade Project no. CUC18A016-1; sponsored by Project no. 2018XNG1857; sponsored by Project no. CUC18QB46.

References

[1] F. A. Gers and J. Schmidhuber, "Recurrent nets that time and count," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN '00)*, vol. 3, p. 6, Como, Italy, July 2000.

[2] K. Cho, B. van Merrienboer, C. Gulcehre et al., "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, October 2014.

[3] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning (ICML '15)*, pp. 448–456, July 2015.

[4] S. Hershey, S. Chaudhuri, D. P. Ellis et al., "CNN architectures for large-scale audio classification," in *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 131–135, New Orleans, LA, March 2017.