ACOUSTIC SCENE GENERATION WITH CONDITIONAL SAMPLERNN

Qiuqiang Kong[1]*, Yong Xu*[2]*, Turab Iqbal*[1]*, Yin Cao*[1]*, Wenwu Wang*[1]*, Mark D. Plumbley*[1]

[1] Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, UK
[2] Tencent AI lab, Bellevue, USA
{q.kong, t.iqbal, yin.cao, w.wang, m.plumbley}@surrey.ac.uk

ABSTRACT

Acoustic scene generation (ASG) is a task to generate waveforms for acoustic scenes. ASG can be used to generate audio scenes for movies and computer games. Recently, neural networks such as SampleRNN have been used for speech and music generation. However, ASG is more challenging due to its wide variety. In addition, evaluating a generative model is also difficult. In this paper, we propose to use a conditional SampleRNN model to generate acoustic scenes conditioned on the input classes. We also propose objective criteria to evaluate the quality and diversity of the generated samples based on classification accuracy. The experiments on the DCASE 2016 Task 1 acoustic scene data show that with the generated audio samples, a classification accuracy of 55.5% can be achieved compared to samples generated by a random model of 6.7% and samples from real recording of 83.1%. The performance of a classifier trained only on generated samples achieves an accuracy of 51.3%, as opposed to an accuracy of 6.7% with samples generated by a random model.

***Index Terms*—** acoustic scene generation, SampleRNN, recurrent neural network, generative model

1. INTRODUCTION

An acoustic scene is a formation of sounds that characterizes a particular place. For instance, the sound of a train stopping could indicate a train station, and the sound of typing could mean that it is an office environment. Often, an acoustic scene is identified by the combination of several different sounds. Not only does it depend on what the sound sources are, but also on characteristics such as loudness and reverberation.

Generating acoustic scenes has a number of applications, such as sound production for movies and computer games [1]. Despite the availability of several datasets for acoustic scenes, the datasets are usually only of hours in total length [2, 3, 4]. Generating acoustic scenes would help to expand the datasets. Similarly, one could use the additional data for other audio classification tasks by incorporating the generated audio with other sounds. For example, adding a background scene to speech could provide a greater diversity of examples for speech recognition. This would allow content creators to include more variations of scenes without recording all of them.

Recently, neural network methods such as WaveNet [5] and SampleRNN [6] have been used to generate raw waveforms. WaveNet and SampleRNN are autoregressive models. WaveNet is based on a convolutional neural network (CNN) that is the audio equivalent of the PixelCNN architecture used in computer vision [7, 8]. SampleRNN is based on a number of recurrent neural networks (RNNs) that correspond to a hierarchy of different temporal resolutions. The fact that it explicitly captures multiple resolutions gives SampleRNN an advantage in generation time over WaveNet in modeling music and speech [6]. Other generative models include variational autoencoders (VAEs) [9] and generative adversarial networks (GANs) [10]. Conditional SampleRNN is used in speech recognition in [11]. Other works for scene generation includes [12]. However, there is not much work in generating a variety of acoustic scenes.

Evaluation of a generative model is important in evaluating the quality and diversity of the generated samples. Previous work used the likelihood of the generated samples on evaluation data to evaluate the generation quality [5]. However, likelihood is often not positively related with the generation quality [13]. For example, a successfully generated white noise acoustic scene has high entropy and low likelihood. Subjective preference scores are widely used to evaluate generation quality [5], but they are time-consuming to obtain and the results may not be reproducible. Recently, objective criteria such as inception scores [14] have been used to evaluate the quality and diversity of generated samples. However, inception scores only evaluate the class diversity of the generated samples and not the diversity of samples within a certain class.

In this paper, we propose to use a conditional SampleRNN model to generate acoustic scenes conditioned on different acoustic classes. In addition, we propose an evaluation criterion inspired by inception scores to evaluate both the quality and variability of the generated samples. This paper is organized as follows: Section 2 introduces the conditional SampleRNN method for generating acoustic scenes. Section 3 introduces the proposed evaluation metric and the proposed evaluation criteria. Section 4 presents experimental results. Section 5 concludes and forecasts future work.

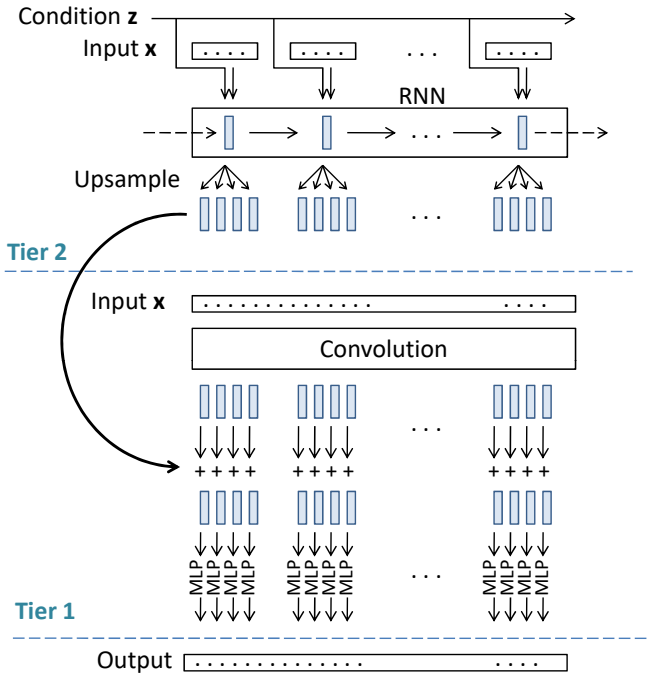

Fig. 1. A two-tier conditional SampleRNN model.

2. AUDIO GENERATION MODEL

Generative models including WaveNet [5] and SampleRNN [6] generate the probability of a sequence of samples $X = \{x_1, x_2, ..., x_T\}$ as the product of the probabilities of each sample conditioned on all previous samples:

$$p(X) = \prod_{t=0}^{T-1} p(x_{t+1}|x_1, ..., x_t). \quad (1)$$

Modeling Equation (1) is difficult when the sequence length T is long. WaveNet [5] models Equation (1) by a stack of dilated convolutional layers. The model outputs the value probability of the next sample. SampleRNN [6] combines the long time and short time dependency information of samples and shows better generation quality of music and speech than WaveNet [6]. In this paper we adopt SampleRNN as our basic model.

2.1. Conditional SampleRNN

SampleRNN consists of several tiers to extract information from different levels [6]. Fig. 1 shows the framework of a two-tier SampleRNN. On the top tier (Tier 2), the audio samples are split into non-overlapping frames. These frames are input to a RNN to learn long dependency information from previous samples. Compared with applying a conventional RNN on the waveform samples directly, SampleRNN reduces the depth of a conventional RNN by a factor of the frame size (shown in Fig. 1). Thus SampleRNN is easier and faster to train [6] because there are fewer temporal steps. The outputs of the RNN in Tier 2 are upsampled by a factor of frame and combine with the information from Tier 1. In Tier 1, audio samples are input to a convolutional layer to capture the short time dependency of adjacent samples. The output of the convolutional layer is added with the output of Tier 1 to utilize both long time and short time information. Finally, a multilayer perceptron (MLP) with fully connected layers is applied to predict the probability of the output audio samples in sample level (Fig. 1).

A conventional SampleRNN is not conditioned on any class. For acoustic scene generation, the aim is to design a generative model that is able to generate audio samples for different acoustic scenes. We propose to use a conditional SampleRNN [11] to generate a variety acoustic scenes. We encode the class information to one-hot encoding $z = \{0, 1\}^K$, where K is the number of acoustic scene classes. Then the one-hot encoding is copied to frames as additional information to the input of the sampleRNN model (Fig. 1).

The mathematical formulation of a conditional two tiers sampleRNN is defined as follows. We use t and j to denote the index of waveform samples and frames, respectively. In Tier 2, an input sequence X is split into non-overlapped frames $X^j = \{x_{(j-1)\times M+1\,:\,j\times M}\}$, where M is the frame size. The two tiers SampleRNN model can be written as:

$$\begin{aligned} u^j &= f_{\mathrm{R}}(WX^j + Vz + b) \\ v^j &= f_{\mathrm{U}}(u^j) \\ q &= \phi(Q * x + c) \\ y^t &= f_{\mathrm{mlp}}(v^t + q^t). \end{aligned} \quad (2)$$

The function f_{R} denotes the recurrent connection in Tier 2 to capture the long time dependency between frames. Symbols $W \in \mathbb{R}^{H\times M}$ and $V \in \mathbb{R}^{H\times K}$ are embedding mappings for the input and conditional class, where H denotes the number of hidden units. The function f_{U} upsamples the output of the recurrent layer in Tier 2 by a factor of the frame size M. The third line of Equation (2) corresponds to the convolution of the input samples in Tier 1, where $Q \in \mathbb{R}^{H\times M}$ is a convolution kernel to model the short time dependency of adjacent M samples. The output from Tier 2 and Tier 1 are summed followed by a multilayer perceptron f_{mlp} with fully connected layers to predict the output samples. In generation, by initializing the starting input sequence as 0 and conditioned on one-hot class condition z, a generated waveform x_{gen} will be obtained by applying equation (2) recursively. We denote the generation function as $x_{\mathrm{gen}} = g(z)$.

3. EVALUATION

Evaluating a generative model is not trivial because the training loss is not usually related with the generation quality [13]. For example, a successfully generated white noise acoustic scene has high entropy and low likelihood. Subjective preference scores are used to evaluate speech generation quality in [5, 6] but they are time consuming to obtain and the result may not

be reproducible. Inception score [14] is an objective criterion defined as $\exp(\mathbb{E}_x[KL(p(y|x) \parallel p(y))])$, where $p(\cdot)$ is a pre-trained classifier. The term $p(y|x)$ indicates generation quality. The term $p(y)$ indicates the generation diversity among classes but does not indicate the diversity in a given class. Therefore inception score is not suitable for evaluating the diversity of generated samples conditioned on a specific class.

3.1. Generation quality

Similar to the inception score, we start with training a classifier f_{real} on a set of real acoustic scene data x_{real}. Then the trained classifier f_{real} is used to classify a set of the generated data $x_{\text{gen}} = g(z)$. As f_{real} is trained on real data, thus it is able to distinguish different sound classes. If the generated samples are of high quality, then $f_{\text{real}}(x_{\text{gen}})$ will have high accuracy in predicting x_{gen} as class z. If the generated samples are of low quality such as random noise, f_{real} tends to predict x_{gen} as a random class.

3.2. Generation diversity

To evaluate the intra-class generation diversity, we train a classifier f_{gen} on a set of the generated data $x_{\text{gen}} = g(z)$ where the set contains generated samples conditioned on all classes. Then the trained f_{gen} is used to classify the real data x_{real}. If the mode of the generated samples collapses, that is, there is little diversity of the generated samples then f_{gen} will have low classification accuracy on x_{real}. On the other hand, if the generation diversity is high and the generated data distribution is close to the real data distribution then the classification performance of f_{gen} should approach f_{real}.

4. EXPERIMENTS

4.1. Dataset

We evaluate the proposed conditional SampleRNN generative model on the DCASE 2016 Task 1 acoustic scene dataset consists of 15 acoustic scenes. For each acoustic scene, there are 39 minutes and 13 minutes audio recordings for training and evaluation. Each audio recording has a duration of 30 seconds. Following [6], we split the training audio recordings into 8-second audio clips. Each 8-second audio clip inherits the label from the original 30-second audio recording. This results in 5728 8-second audio clips for training, where 512 8-second audio clips are held out for validation. The generated samples and source code are available on GitHub[1].

4.2. Model

A two-tier conditional SampleRNN is applied as the generative model (Fig. 1). In Tier 2, we set the frame size to 16 and the number of frames to 64, resulting in 1024 samples used for back propagation through time (BPTT). Long time dependency information is used to initialize the states of the RNN in Tier 2. We model the RNN with a three-layer gated recurrent unit (GRU) [15] with 1024 hidden units. The upsampling layer has an upsample factor of 16, which is the same as the frame size. In Tier 1, the frame size of the receptive field of the convolutional layer is 16. The output of the convolutional layer and the output of Tier 2 are added, followed by a multilayer perceptron with two fully connected layers with 1024 hidden units to predict the output samples. During training, the Adam optimizer [16] with a learning rate of 0.001 is used. The model is trained for 200,000 iterations, which takes two days on a TITAN XP single-GPU card.

We apply a 4-layer CNN as the classification model [17]. The CNN consists of four layers with 32, 64, 128 and 256 feature maps in each layer. Global average pooling is applied on the final feature map followed by a fully-connected layer with a softmax nonlinearity to predict the presence probability of acoustic scene classes. A dropout [18] rate of 0.5 is applied after every convolutional layer to prevent the system from

[1] https://github.com/qiuqiangkong/sampleRNN_acoustic_scene_generation

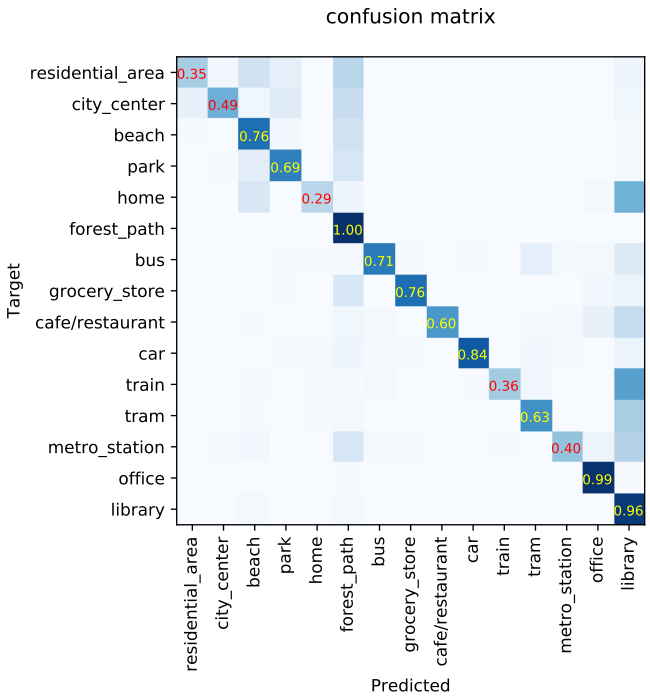

Fig. 2. Confusion matrix of the classification accuracy on the generated audio samples.

Table 1. Classification accuracy on the evaluation, conditional SampleRNN generated and randomly generated waveforms.

	Evaluation	Generated	Random
Accuracy	83.1%	65.5%	6.7%

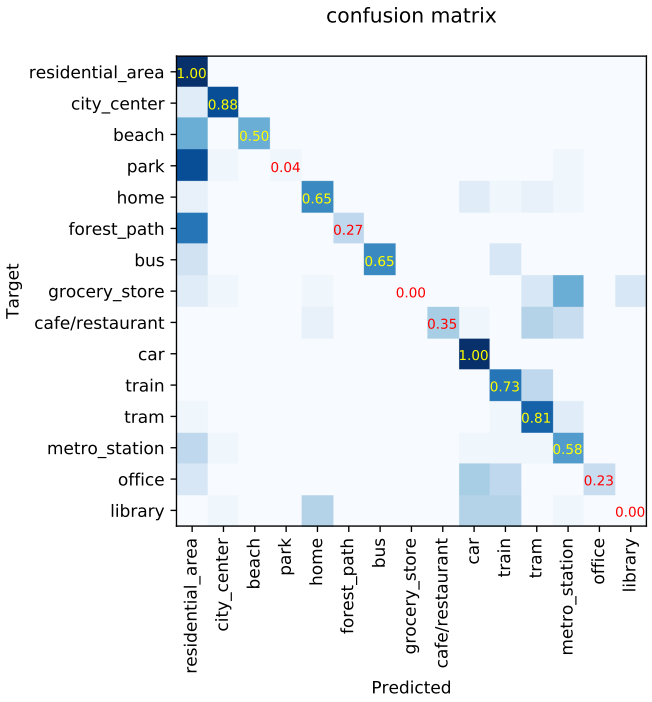

Fig. 3. Classification accuracy on the evaluation data using the classification model trained on generated samples only.

overfitting. Similar to the generative conditional SampleRNN model, the classification model is trained using Adam with a learning rate of 0.001 following [17].

4.3. Generation quality

We generate 2,000 audio samples for each acoustic scene, resulting in 30,000 generated audio samples in total. Table 1 shows that using the classifier f_{real} trained on the training data a classification accuracy of 83.1%, 65.5% and 6.7% is obtained on the evaluation set, conditional SampleRNN generated waveforms and random waveforms, respectively. Fig. 2 shows the confusion matrix of the classification result on the generated audio samples. Audio classes such as "car" and "office" have good generation quality. The results indicate that a majority of the generated audio waveforms are indistinguishable from the real audio waveforms because f_{real} classifies these generated waveforms correctly. Compared with acoustic scene generation with replaying the training waveforms, the SampleRNN can generate infinite acoustic scenes.

4.4. Generation diversity

A good generative model should have diversity of the generated waveforms [14], i.e. the distribution of the generated data should be close to that of the real data, and should not collapse to a single mode. To evaluate the generation diversity, we train classification models f_{gen} on the generated samples only. Then,

Table 2. Classification accuracy on the training and evaluation audio samples using the f_{gen}.

Gen. samples	Train	Evaluate
1	0.176	0.146
2	0.296	0.069
5	0.433	0.177
10	0.483	0.246
20	0.563	0.274
50	0.590	0.308
100	0.617	0.249
200	0.668	0.397
500	0.688	0.487
1000	**0.694**	0.495
2000	0.688	**0.513**

f_{gen} is used to classify the training and evaluation waveforms. In our experiments, the number of generated samples per class is ranged from 1 to 2000 for training f_{gen}. Table 2 shows the classification accuracy on the training and evaluation data with f_{gen} trained on different number of generated samples. The classification accuracy on the training data increases from 0.176 to 0.694 with 1 to 1000 generated audio samples per class for training. The classification accuracy on the evaluation data increases from 0.146 to 0.513 with 1 to 2000 generated audio samples per class for training. Fig. 3 shows the confusion matrix of the accuracy of f_{gen} on the evaluation data. Classes such as "beach" and "car" have high classification accuracy indicating their high generation diversity.

5. CONCLUSION

We have presented a conditional SampleRNN model for generating waveforms for acoustic scenes. The generative model can be conditioned on different acoustic scenes. We propose to evaluate the generation quality using a classifier trained on the real waveforms and evaluate the generation diversity using a classifier trained on generated waveforms. Using the classifier trained on real waveforms, an accuracy of 65.5% is achieved on the generated waveforms, indicating the high quality of the generated samples. Using the classifier trained on generated waveforms, the performance improves with the number of generated waveforms. This indicates that the generated waveforms are different thus have a good diversity. In future, we will investigate acoustic scene generation with more autoregressive models.

6. ACKNOWLEDGEMENT

This research was supported by EPSRC grant EP/N014111/1 "Making Sense of Sounds" and a Research Scholarship from the China Scholarship Council (CSC) No. 201406150082.

7. REFERENCES

[1] D. B. Lloyd, N. Raghuvanshi, and N. K Govindaraju, "Sound synthesis for impact sounds in video games," in *Symposium on Interactive 3D Graphics and Games*. ACM, 2011, pp. 55–61.

[2] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *European Signal Processing Conference*, 2016, pp. 1128–1132.

[3] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *ACM International Conference on Multimedia*, New York, 2015, pp. 1015–1018.

[4] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *ACM International Conference on Multimedia*, New York, 2014, pp. 1041–1044.

[5] A. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," in *Speech Synthesis Workshop*, 2016, p. 125.

[6] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, "SampleRNN: An unconditional end-to-end neural audio generation model," *International Conference for Learning Representations (ICLR)*, 2017.

[7] A. Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *Proceedings of Machine Learning Research*, New York, 2016, vol. 48, pp. 1747–1756.

[8] A. Aäron van den, N. Kalchbrenner, L. Espeholt, K. kavukcuoglu, O. Vinyals, and A. Graves, "Conditional image generation with PixelCNN decoders," in *Advances in Neural Information Processing Systems*, Barcelona, Spain, 2016, pp. 4790–4798.

[9] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," *Journal of Machine Learning Research*, pp. 1929–1958, 2014.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[11] C. Zhou, M. Horgan, V. Kumar, C. Vasco, and D. Darcy, "Voice conversion with conditional SampleRNN," *arXiv preprint arXiv:1808.08311*, 2018.

[12] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2017, pp. 344–348.

[13] L. Theis, A. Oord, and M. Bethge, "A note on the evaluation of generative models," *International Conference for Learning Representations*, 2016.

[14] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Advances in Neural Information Processing Systems*, 2016, pp. 2234–2242.

[15] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference for Learning Representations (ICLR)*, 2015.

[17] Q. Kong, T. Iqbal, Y. Xu, W. Wang, and M. D. Plumbley, "DCASE 2018 Challenge baseline with convolutional neural networks," in *DCASE Workshop*, 2018.

[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.