Conditional WaveGAN

Chae Young Lee *
HAFS
cylee@hafs.hs.kr

Anoop Toffy *
IIIT Bangalore
anoop.toffy@iiitb.org

Gue Jun Jung
SK Telecom
guejun.jung@sk.com

Woo-Jin Han
Netmarble IGS
wjhan@igsinc.co.kr

Abstract

Generative models are successfully used for image synthesis [11] in the recent years. But when it comes to other modalities like audio, text etc little progress has been made. Recent works focus on generating audio from a generative model in an unsupervised setting. We explore the possibility of using generative models conditioned on class labels. Concatenation based conditioning and conditional scaling were explored in this work with various hyper-parameter tuning methods [22] [15]. In this paper we introduce Conditional WaveGANs (cWaveGAN). Find our implementation at https://github.com/acheketa/cwavegan

1 Introduction

Generative adversarial networks [8] are being widely used for synthesizing realistic images [13, 20, 1]. But very little has been explored in the area of audio generation. A few research works has been made in the area of unsupervised generative models in audio. One of them is WaveGAN [5], which trains a generative model in an unsupervised setting. In this work we use WaveGAN model as our baseline model. Audio samples generated from WaveGAN are human-recognizable and have a relatively good inception scores [2]. But the samples generated are completely random.

The need for generating synthetic data has many applications in various fields. For example, in digital image enhancement, synthesizing new images can be used for smoothing (denoising), filling missing pieces (inpainting), improving resolution (super-resolution imaging) etc. Other areas include reinforcement learning, where agents can be trained with more training examples, and automatic speech recognition, where the synthesized data from generative models can be used as training data for training acoustic models. More direct usage is in the field of semi-supervised learning techniques [23, 18, 22]. In the recent 2018 ACM webinar [12] Ian Goodfellow talks about the semi-supervised learning techniques and their applications in various fields.

In this work we explore a way to generate audio samples conditioned on class labels. That is, given a class label whether the generator of GAN can generate a particular audio waveform. In the history of GAN such type of conditioning has been explored in the synthesis of images. Conditional Generative Adversarial Nets introduced in 2016 by Mehdi Mirza et. al [17] was the first to try concatenation based conditioning. The conditional GANs were able to synthesize realistic images on MNIST and MIR Flickr datasets.

* These two authors contributed equally

Preprint. Work in progress.

2 Related Work

Ever since the introduction of Generative Adversarial Networks by Ian Goodfellow et. al. [8] in 2014 a lot of varieties of GANs has been implemented for accomplishing various tasks. It is one of the much explored generative models when compared to Variational Autoencoders (VAEs) and autoregressive models, assisted by a great tutorial given by Ian Goodfellow in NIPS 2016 [7]. Most of the notable advancements were made in the field of image synthesis. In his 2018 paper, He Huang et al. [11] talks more about the usage of GANs for image synthesis and various techniques in this fields.

Usage of generative models in natural language processing is also in progress with notabld fields including text modelling [28, 21], dialogue generation [16], and neural machine translation [27]. When it comes to audio generation, several approaches have been explored. Nonetheless, models that can generate audio waveforms directly (as opposed to some other representation that can be converted into audio afterwards, such as spectrograms or piano rolls) are only starting to be explored. Autoregressive models were initially used for the generation of raw audio. WaveNet by Van Den Oord et al. [25] is a convolutional model with dilated convolutions that learns to generate raw audio by autoregressive modelling. While a single WaveNet can capture the characteristics of many different speakers with equal fidelity and can switch between them by conditioning on the speaker identity, WaveRNN [14] describes a single-layer recurrent neural network with a dual softmax layer that matches the quality of the state-of-the-art WaveNet model. WaveNets have been applied to music generation as well. The paper by Sander Dieleman et al. (2018) [4] talks about the challenges in realistic music generation.

The application of GAN in speech were used mainly for audio enhancements techniques and little on raw audio generation. SEGAN by Santiago Pascual et al [19] uses deep networks which operates on waveform level, training the model end-to-end for denoising waveform chunks. The model works as a fully convolutional encoder-decoder structure. Tacotron [26] is an end-to-end generative text-to-speech model that synthesizes speech directly from characters. Since Tacotron generates speech at the frame level, it's substantially faster than sample-level autoregressive methods. In Tacotron2 [24] a neural network architecture synthesizes speech directly from the text. Tacotron2 achieves a mean opinion score (MOS) of 4.53, whereas a professionally recorded speech obtained a MOS of 4.58.

In the recent work on speech synthesis, Chris Donahue et al. (2018) [5] introduces two GAN models, WaveGAN and SpecGAN. The waveGAN works in the time-domain and specGAN works in the frequency-domain. WaveGAN can produce intelligible words from a small vocabulary of human speech, as well as synthesize audio from other domains such as bird vocalizations, drums, and piano. It uses GANs in an unsupervised setting. WaveGAN is based on the DCGAN [20] architecture, which became famous by its usage in image synthesis. We focus on this work to make the WaveGAN conditioned on class labels by introducing various conditioning techniques discussed in the following sections.

3 Adversarial Nets

3.1 Generative Adversarial Nets

GAN was introduced as a novel method to train generative models. They are composed of two adversarial models: the *generator, G* and the *discriminator, D*. The generator, G captures the data distribution while the discriminator, D estimates the probability that the data generated by generator is coming from the data distribution or not. Both G and D can be non-linear mapping functions.

To learn the data distribution P_D over data x, the generator builds a mapping from a noise distribution P_z to data space as $G(z, \theta_g)$. And the discriminator $D(x, \theta_d)$ outputs as scalar representing that the training data p_x rather than p_g.

G and D are both trained simultaneously: we adjust parameters for G to minimize $log(1 - D(G(z))$ and adjust parameters for D to minimize $logD(X)$, as if they are following the two-player min-max game with value function $V(G, D)$:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]. \quad (1)$$

3.2 Conditional Generative Adversarial Nets

The conditional GAN (cGANs) [17] were introduced to make use of the additional information available in the form of labels, y. So instead of generating a random data from generator the auxiliary information like labels, y are passed to both generator and discriminator along with input features, (z and x) so as to generate data conditioned on a class label.

In both generator and discriminator the label information, y is augmented with the input features (z and x).

The objective function of a two-player minimax game would be as Eq. 2.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x|y)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z|y)))]. \tag{2}$$

4 Conditioning methodologies

In this work we explore a few conditioning mechanisms discussed in Dumoulin et. al [6] article in this section. We are interested in generating raw audio of a particular given class label from the generator output. That is, the model takes as input a class and a source of random noise (e.g., a vector sampled from a normal distribution) and outputs a raw audio sample for the requested class as shown in Fig. 1.

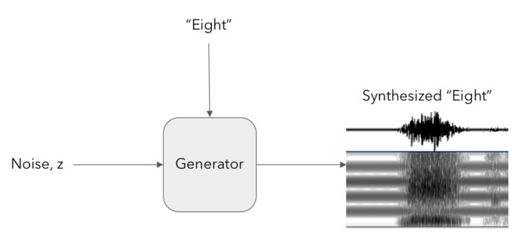

Figure 1: Synthesizing raw audio by conditioning

4.1 Concatenation based conditioning

First approach is to embed class information to input feature vector. That is, before training we would concatenate a representation of conditioning information to the noise vector and use it as the model's input as shown in the figure 2

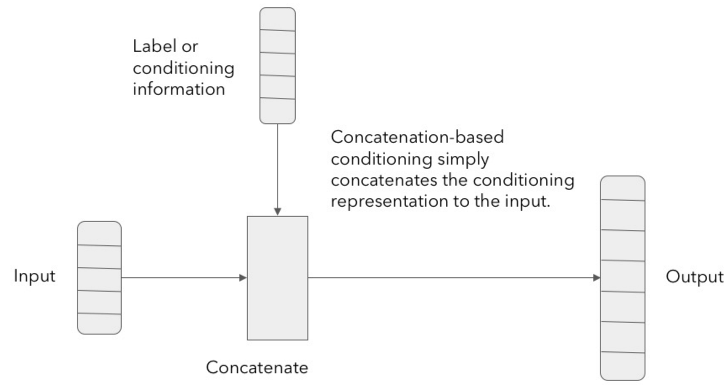

Figure 2: Concatenation based conditioning

4.2 Conditional scaling

In the other type of conditioning we implemented we scaled the hidden layers based on the conditioning representation and multiplied it with the input vector for both discriminator and generator input. Also the scaling is applied to each layer of the convolution model.

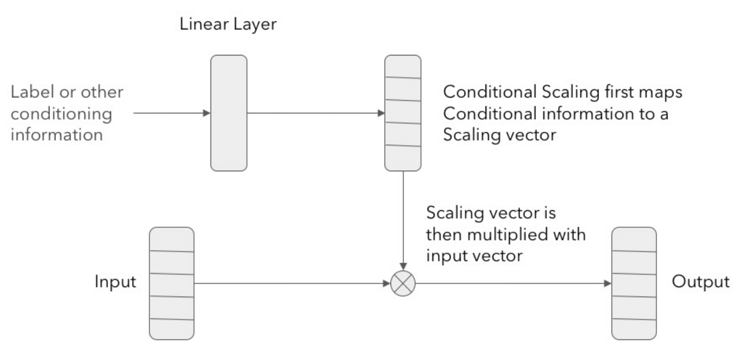

Figure 3: Conditional scaling

5 Experimental Setting

We based our architecture on WaveGAN model used by Chris Donahue et. al. [5] which inturn is based on the DCGAN model by Radford et. al [20]. We modified the input vector dimension to 8192 by removing the beginning and ending silence to better capture the word alone. As in WaveGAN paper we used one-dimensional filters of length 25 instead of two-dimensional filters of size 5x5 used in DCGAN model architecture but we changed the strides from 4 to 2 just to incorporate the change in input vector dimension. The architecture of generator and discriminator model is provided in Table 1 and Table 2 respectively.

Table 1: Conditional WaveGAN generator architecture

Operation	Kernel Size	Output Shape
Input $z \sim Uniform(-1, 1)$		(n, 100)
Dense 1	(100, 256d)	(n, 256d)
Reshape		(n, 16, 16d)
ReLU		(n, 16, 16d)
Trans Conv1D (Stride=4)	(25, 16d, 8d)	(n, 64, 8d)
ReLU		(n, 64, 8d)
Trans Conv1D (Stride=4)	(25, 8d, 4d)	(n, 256, 4d)
ReLU		(n, 256, 4d)
Trans Conv1D (Stride=4)	(25, 4d, 2d)	(n, 1024, 2d)
ReLU		(n, 1024, 2d)
Trans Conv1D (Stride=4)	(25, 2d, d)	(n, 4096, d)
ReLU		(n, 4096, d)
Trans Conv1D (Stride=2)	(25, d, c)	(n, 8192, c)
Tanh		(n, 8192, c)

Table 2: Conditional WaveGAN discriminator architecture

Operation	Kernel Size	Output Shape
Input x or $G(z)$		(n, 8192, c)
Conv1D (Stride=2)	(25, c, d)	(n, 4096, d)
LReLU ($\alpha = 0.2$)		(n, 4096, d)
Phase Shuffle ($n = 2$)		(n, 4096, d)
Conv1D (Stride=4)	(25, d, 2d)	(n, 1024, 2d)
LReLU ($\alpha = 0.2$)		(n, 1024, 2d)
Phase Shuffle ($n = 2$)		(n, 1024, 2d)
Conv1D (Stride=4)	(25, 2d, 4d)	(n, 256, 4d)
LReLU ($\alpha = 0.2$)		(n, 256, 4d)
Phase Shuffle ($n = 2$)		(n, 256, 4d)
Conv1D (Stride=4)	(25, 4d, 8d)	(n, 64, 8d)
LReLU ($\alpha = 0.2$)		(n, 64, 8d)
Phase Shuffle ($n = 2$)		(n, 64, 8d)
Conv1D (Stride=4)	(25, 8d, 16d)	(n, 16, 16d)
LReLU ($\alpha = 0.2$)		(n, 16, 16d)
Reshape		(n, 256d)
Dense	(256d, 1)	(n, 1)

Table 3: Conditional WaveGAN hyperparameters

Name	Value (GPU)	Value (GPU)	Value (TPU)	Value (TPU)
Input data type	16-bit PCM	16-bit PCM	16-bit PCM	16-bit PCM
Model data type	32-bit float	32-bit float	32-bit float	32-bit float
Num channels (c)	1	1	1	1
Batch size (b)	64	64	1024	1024
Model size (d)	64	64	64	64
Phase shuffle (WaveGAN)	2	2	2	2
Loss	WGAN-GP [10]	DCGAN [20]	WGAP-GP	DCGAN
D updates per G update	5	5	5	5
Optimizer	Adam ($\alpha = 1\text{e}{-4}$)	Adam ($\alpha = 2\text{e}{-4}$)	Adam ($\alpha = 2\text{e}{-4}$)	Adam ($\alpha = 2\text{e}{-4}$)

As a pilot study for this project we trained our model with DCGAN [20] and WAGAN-GP [10] losses with and without batch normalization. We compare two combinations in the following table 3. We tried with different learning rate apart from $\alpha = 2\text{e}{-4}$, ie with $\alpha = 2\text{e}{-2}$ and $\alpha = 4\text{e}{-4}$. We also tried different learning rates with generator and disriminator since we found discriminator loss graph spiking.

We train our networks using batches of size 64 on four NVIDIA Telsa V100 accelerators and batches of 1024 on a single 8 core TPUv2. We trained it over 200 epochs for around 3 days.

6 Dataset

We used the *Speech Commands Dataset* [3] released by Google AI Team for conducting our prilimi-nary experiments. This dataset consists of many speakers recording individual words in uncontrolled recording conditions. We used a subset of the dataset as done by the authors of WaveGAN paper, ie. Speech Commands Zero Through Nine (SC09) subset, which reduces the vocabulary of the dataset to ten words: the digits “zero” through “nine”.

Each recording utterance is of one second in length and the training set consists of 1850 recording for each word adding upto 5.3hr of speech data. The wide variety of alignments, speakers and recording conditions make this a challenging dataset from a generative perspective.

7 Preliminary Results

We were able to generated human realistic raw audio from the conditional WaveGAN. But most of the time we faced the difficulty of having generated noisy output. The conditioning that we tried we not consistent more often we ended up generating a false positive sample. So we will be research further of stabilizing GAN training and tuning our training parameters to be generate raw audio out that are realistic to real world data.

This is a pilot study which explores the possibility of making use of synthesized data from generative models to be used as training data by other discriminative models.

8 Conclusion

Synthesizing audio is a area that has been growing recently and is much focused by researcher since it has its application in building robust speech recognition systems and improved text-to-speech systems. Much research were focused on generation on random audio from unlabeled training data. In this work we focused on generation of audio given a class label. We were able to get human recognizable synthesized audio waveform from conditional waveGANs, but the presence of noise in the sample generated needs to be smoothed as a continuation of the present work, which gives a lot of scope for future improvements. Secondly, hyperparameter tuning of GAN needs to be explored since presently the training is unstable. Thirdly, the conditioning method explored so far is increasing training time of GANs, so a more novel conditioning method need to be discovered.

We hope this work gives direction to more research on conditional generation of audio wave-forms and gets extended to other frontiers.

Acknowledgments

We would like thank Tensorflow Korea and Google Korea, and all other sponsors of Deep Learning Camp Jeju 2018, `http://jeju.dlcamp.org/2018/`. This work was done as a part of the DL Jeju Camp. We would like to convey our sincere gratitude to our mentors, Dr.Gue Jun Jung (SK Telecom), Dr. Woo-Jin Han (Netmarble IGS) and all other men-tees of the camp.

We are also delighted to use TPUs [9] for our training process which made a considerable improvement in making the training process faster.

References

[1] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. Van Gool. Generative adversarial networks for extreme learned image compression. *arXiv preprint arXiv:1804.02958*, 2018.

[2] S. Barratt and R. Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

[3] J. Buchner. Synthetic speech commands dataset. *https://www.kaggle.com/jbuchner/synthetic-speech-commands-dataset*, 2017.

[4] S. Dieleman, A. v. d. Oord, and K. Simonyan. The challenge of realistic music generation: modelling raw audio at scale. *arXiv preprint arXiv:1806.10474*, 2018.

[5] C. Donahue, J. McAuley, and M. Puckette. Synthesizing audio with generative adversarial networks. *arXiv preprint arXiv:1802.04208*, 2018.

[6] V. Dumoulin, E. Perez, N. Schucher, F. Strub, H. d. Vries, A. Courville, and Y. Bengio. Feature-wise transformations. *Distill*, 3(7):e11, 2018.

[7] I. Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. URL `http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf`.

[9] Google. Cloud tpu. *https://cloud.google.com/tpu/*, 2016.

[10] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[11] H. Huang, P. S. Yu, and C. Wang. An introduction to image synthesis with generative adversarial nets. *arXiv preprint arXiv:1803.04469*, 2018.

[12] G. B. Ian Goodfellow, Staff Research Scientist. Adversarial machine learning. *https://goo.gl/NVWF7j*, 2016.

[13] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

[14] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. v. d. Oord, S. Dieleman, and K. Kavukcuoglu. Efficient neural audio synthesis. *arXiv preprint arXiv:1802.08435*, 2018.

[15] K. Kurach, M. Lucic, X. Zhai, M. Michalski, and S. Gelly. The gan landscape: Losses, architectures, regularization, and normalization. *arXiv preprint arXiv:1807.04720*, 2018.

[16] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*, 2017.

[17] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[18] A. Odena. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016.

[19] S. Pascual, A. Bonafonte, and J. Serra. Segan: Speech enhancement generative adversarial network. *arXiv preprint arXiv:1703.09452*, 2017.

[20] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[21] S. Rajeswar, S. Subramanian, F. Dutil, C. Pal, and A. Courville. Adversarial generation of natural language. *arXiv preprint arXiv:1705.10929*, 2017.

[22] 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*, pages 2234–2242, 2016.

[23] L. Schoneveld. Semi-supervised learning with generative adversarial networks. 2017.

[24] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. *arXiv preprint arXiv:1712.05884*, 2017.

[25] A. Van Den 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 *SSW*, page 125, 2016.

[26] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, et al. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*, 2017.

[27] Z. Yang, W. Chen, F. Wang, and B. Xu. Improving neural machine translation with conditional sequence generative adversarial nets. *arXiv preprint arXiv:1703.04887*, 2017.

[28] L. Yu, W. Zhang, J. Wang, and Y. Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, pages 2852–2858, 2017.