JINGLEBOT: RAW-AUDIO GENERATION WITH WAVENET

Hanna Berger
University of Rochester
hberger6@u.rochester.edu

Anna Boyd
University of Rochester
aboyd13@u.rochester.edu

Izzy Hargrave
University of Rochester
ihargrav@u.rochester.edu

ABSTRACT

We present an in-progress model for short-form music generation based on the WaveNet model for raw audio to generate short-form musical snippets conditioned on emotional categories. We pre-trained our model using the Free Music Archive dataset. Additional work involves fine-tuning the model with an original dataset of jingles labeled by emotional category. The complete model will generate jingles conditioned on mood.

1. INTRODUCTION

Jingles are short musical fragments, usually a simple melody with one or two timbres. They are used often in advertising to create brand awareness by associating a brand with a short, catchy melody. Jingles are also used as music beds underneath dialogue for radio and broadcast. For this experiment, we defined a jingle as a musical fragment of one minute or shorter. The goal of this experiment was to generate jingles of high enough musical quality that they could plausibly be written by a human. To be classified as "music," the jingles should contain recognizable features of harmony, timbre, and rhythmic pattern. Short-form music generation could prove a useful tool for commercial songwriters to generate new musical ideas.

2. BACKGROUND

To accomplish our proposed task, we used WaveNet, an audio generation architecture developed by Google Deepmind in 2016 [1]. WaveNet was originally designed to generate human speech from text, though it has since been employed in a number of music generation tasks [2-4]. At the time of WaveNet's release, most DNN audio generation architectures relied on representations of music such as MIDI or notated scores [5]. More advanced models relied on the extraction of spectrograms, MFCCs, or other features to model complex temporal features. WaveNet was a theoretical breakthrough because it provided a mechanism to directly model raw audio features [6].

WaveNet is primarily used for TTS tasks [1]. We decided to implement a convolutional model based on the WaveNet architecture to further explore its capabilities for music generation. WaveNet has proven to be a robust model for many audio tasks [4-5, 7-9] and we were interested in applying it to jingle generation. A major difficulty with WaveNet is its relatively short receptive field of a few seconds, but jingles are often just a few seconds. All of these reasons made WaveNet an appealing choice for this task.

With current computation limits, it is challenging to model raw audio directly. WaveNet's key contribution was to incorporate the use of dilated convolutions to model long-term temporal relationships without having to process every single data point. Dilated convolutions operate much like a traditional convolution, but they skip input samples at regular intervals. The higher the dilation, the more "stretched out" the convolution is. This approach remains highly successful. At the time of its release, WaveNet offered state-of-the-art performance for text-to-speech tasks. In practice, it had a receptive field of a few seconds; this is usually enough for modeling speech and short musical ideas, like jingles.

3. BUILDING THE WAVENET MODEL

WaveNet is a probabilistic, auto-regressive model designed to predict the next sample in a sequence given all previous samples. The model produces each prediction as a probability distribution over some number of channels, which can then be sampled to generate novel audio. To be mindful of computational restrictions, the 65,536 values of 16-bit audio is typically quantized to 256 channels. The input audio is also typically downsampled to 16 KHz to further reduce the size of the model. The WaveNet architecture as proposed by DeepMind is shown in Figure 1 and further explained in Section 3.1.

3.1 Model Structure

In general, the WaveNet architecture consists of 3 phases. The input audio is first filtered using a causal convolutional layer. In training, the main purpose of this layer is to feed the model previous samples while masking the current sample in the sequence. This ensures the model learns to predict the current sample using only the previous samples.

After the input samples are processed by this initial convolution, they are passed to the body of the model, referred to as the residual stack. This stack is composed of several layers of dilated convolutions which pass the input from one layer to the next in series. A residual is also passed to each block, as seen in Figure 2. This residual propagates gradients further into the network, helping the model to converge [1].

Finally, the output of each layer in the residual stack is summed and passed through two final convolutions and activations. A softmax function is then applied to the output to produce a normalized probability distribution across all channels. During generation, every new sample in a novel sequence is chosen from this distribution.

The performance of the WaveNet model is in part determined by the maximum dilation size of the convolutional layers. This dilation size determines how far back into the signal the model looks when predicting a new sample, as illustrated in Figure 1. This memory is key when attempting to capture and replicate the time dependencies of an audio signal. The success of a model is also predicated on its size and complexity – the number of convolutional layers in the residual stack.

To maximize the performance of our model while working within the limitations of 20GB of GPU RAM, we built our model using a total of 32 dilated convolutional layers with a maximum dilation of 256 samples.

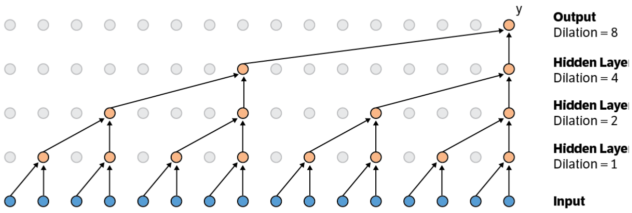

Figure 1. WaveNet Dilated Convolutions [1]

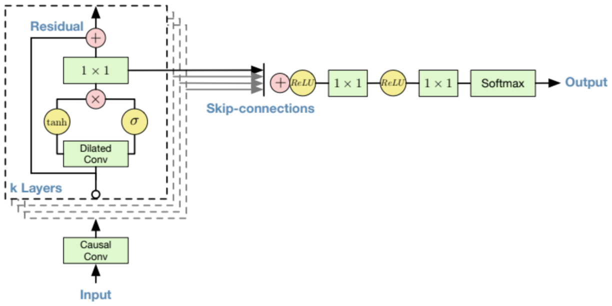

Figure 2. WaveNet Residual Stack Structure [1].

3.2 Generation

To generate novel audio, WaveNet is first fed some seeding audio. This audio may be initialized to zeros, random values, or may be the first several samples of a target audio file. Then, every sample predicted by the model is then fed back into it as a new input. This process is self-perpetuating and can be used to generate audio sequences of any length.

The naive implementation of a generation algorithm is computationally expensive (O(2^{L}) for a model of L layers). This computation can be reduced to O(L) by employing the “Fast WaveNet” generation algorithm proposed by [6]. This algorithm circumvents redundant calculations in the generation process by creating hidden state queues. In this architecture, each layer in the model is given a zero-initialized queue to store its hidden states, as shown in Figure 3. These queues are dynamically updated with each sample.

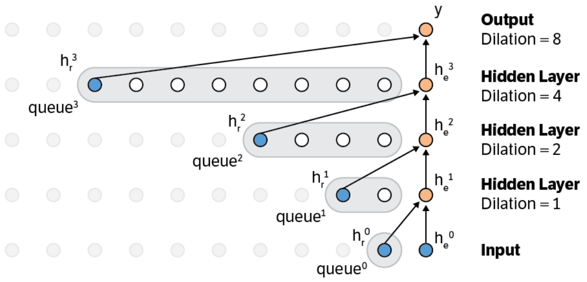

Figure 3. Fast WaveNet Hidden State Queues [6].

3.3 Conditioning

To achieve our goal of mood-based jingle generation, we plan to implement an additional conditioning parameter in the structure of the model. Because the "mood" of a jingle is time-independent, we are able to implement this feature as a form of global conditioning. The "mood" of a jingle is provided to the model as a second input and, over the course of training, the model learns to incorporate this mood into its audio predictions via a linear projection.

This mood feature is propagated through every layer in the residual stack, ensuring that the entire model is informed by the conditioning. This behavior can be seen in Equation 1, which describes the non-linear gating functions found in every convolutional layer of the model. W is the activation matrix for the audio input x, and V is the linear projection matrix for the latent mood h.

$$\mathbf{z} = \tanh\left(W_{f,k} * \mathbf{x} + V_{f,k}^{T}\mathbf{h}\right) \odot \sigma\left(W_{g,k} * \mathbf{x} + V_{g,k}^{T}\mathbf{h}\right). \quad (1)$$

Equation 1. Activation function with global conditioning added from [1].

4. DATASETS

We employed two different datasets in the training of our model. In order to train the WaveNet model to generate raw audio that is coherent and musical, we needed to train the model on a very large set of audio. Because no such dataset exists for jingles, and we had a need for additional training, we decided to train our model in multiple stages using two separate datasets. We first pretrained our model without conditioning on a large dataset of human-generated music. We plan to further train the model with global conditioning on a smaller dataset of jingles.

4.1 The Free Music Archive Dataset

We first trained our model on the Free Music Archive (FMA) dataset [10]. It is made up of 106,574 different royalty free music samples containing different timbres and genres. There are several modified dataset sizes available including Small, Medium, and Large. To begin, we used the FMA Small dataset that contains 8,000 30-second music clips.

4.2 The Jingle Dataset

We also created our own dataset of jingles by pulling from various royalty free music websites [11-15]. We collected 421 jingles in total. We then manually categorized them by listening and assigning them to 5 different moods. We chose these moods based off of what was most commonly heard when listening through the dataset. These moods include upbeat, relaxing, melancholy, mysterious, and playful. Each of the audio files received a label from one of the categories. The moods were abbreviated into three-character-long tags that were added at the beginning of the file names in order to make it easy for the data loader to find them. Our finalized dataset includes 58 melancholy, 53 mysterious, 75 playful, 90 relaxing, and 144 upbeat jingles.

5. EXPERIMENTATION

While building the model, we conducted several experiments to test its functionality. First, we trained the model to closely fit a single song to ensure that it could predict musical sequences. Next, we trained the model on the FMA dataset without conditioning. This served as preliminary training on generic music that taught the model how to predict samples in a musical way. Finally, we plan to further train the model on the Jingle Dataset with global conditioning to teach the model to discriminate between jingles of different moods.

5.1 One Song Experiment

As a proof of concept, we trained the WaveNet model on a single song for several thousand epochs. In theory, by only training the model on one song, the generation algorithm should be able to reconstruct jumbled but highly realistic features of the song.

After choosing a music sample from the FMA Small Dataset, we fit the model to the song, experimenting with different combinations of hyperparameters. As described in Section 3.1, our model contained a total of 32 residual layers with a maximum dilation of 256 samples. Both of the hyperparameter combinations yielded similar results, with neither converging to near-zero (Fig. 4).

Hyperparameters	Trial #1	Trial #2
Batch Size:	1	1
Stack Size:	4	1
Layer Size:	8	24
Learning Rate:	0.003	0.003
Total Epochs Trained:	1252	1536

Table 1. Hyperparameters for the Single Song Trials. Trial 1 has a smaller receptive field, but it incorporates multiple stacks. Trial 2 has a large receptive field, but only has a single stack.

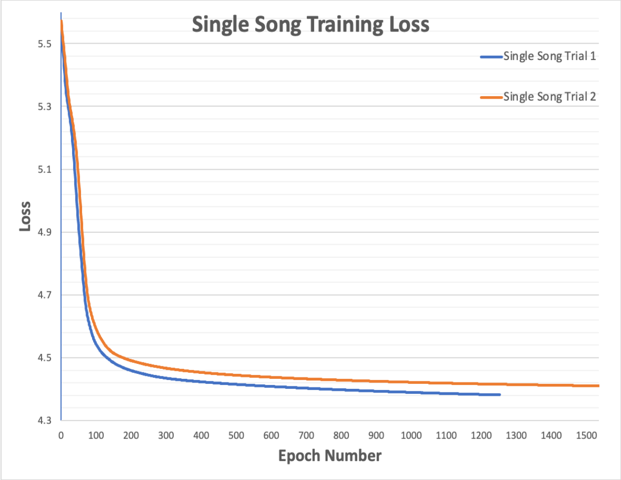

Figure 4. Training Loss from two different trials. With the computation available to us, we found that having at least 4 stacks of residual blocks was worth sacrificing some of the receptive field. Our plan for future research includes additional hyperparameter tuning.

5.2 Model Training

After running the model on one song, we expanded it to the FMA Small Dataset. The training process was very similar to training just one song, but each epoch included 6,400 songs instead of just 1. Although the model showed significant improvement, the loss is still very high. We intend to train the model on the full FMA dataset, which has over a hundred times more data, to combat this issue. Finally, we plan to train our model on the Jingle Dataset with the addition of global conditioning.

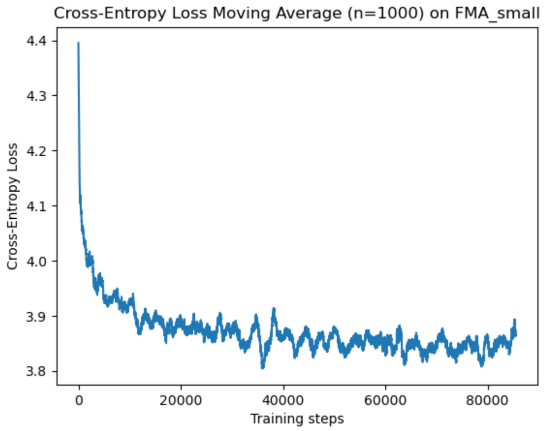

Figure 5. Moving-average training loss for FMA-small. This training job took 96 hours on 2 Nvidia GTX 1080tis.

6. RESULTS

While this model is still a work in progress, we achieved significant results from our WaveNet implementation trained on a single song. A forward pass of the model on an input song yielded a very similar output (Fig. 7) to the input song (Fig. 6). Since the model masks the current sample and uses the ground truth to predict it, this proves that the model is learning and predicting audio samples as intended.

Additionally, we successfully implemented the WaveNet fast generation algorithm [6] and generated noisy but harmonic audio, indicating that the training process simply needs more computation power to learn a more complex representation of the input.

Although our model does not yet generate low-distortion musical fragments, our WaveNet implementation shows promising results. It is able to recognize and replicate simple patterns in raw audio (Fig. 9). The output of the forward pass, which is an audio file of predicted samples (where each sample is predicted by the model given the target audio file) sounds very natural and has a high SDR. The generated audio (where every sample is conditioned on previously generated samples) is much noisier, but audible rhythmic and harmonic patterns are present. This indicates that the model is successful at predicting just a few samples, but less successful when tasked with generating hundreds of thousands of samples conditioned on previous outputs. This is a non-trivial finding: this could be used to up-sample audio files from a smaller sampling rate to a higher one. Preliminary experiments indicated that the model was very good at filling in the gaps between correct samples when it was periodically set back on track by a target audio sample.

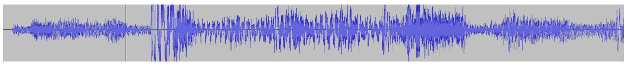

Figure 6. One Song Experiment: Original Audio File

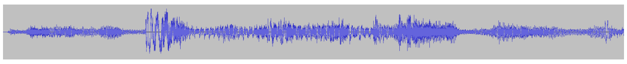

Figure 7. One Song Experiment: Forward Pass

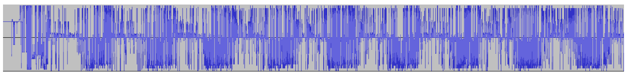

Figure 8. One Song Experiment: Novel Generation

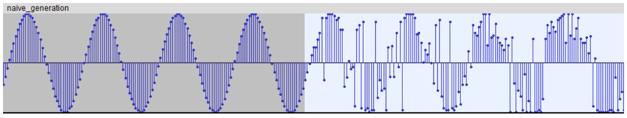

Figure 9. Our model's output for a sine wave.

Name	SDR
Forward Pass	13.596
Generated Audio	-16.825

Table 2. Signal-to-distortion ratios for the forward pass (Fig. 7) and the generated audio (Fig. 8). Higher is better; a positive value means more signal than distortion.

7. DISCUSSION AND FUTURE WORK

A core issue in this research was access to computation. Though the Google Deepmind researchers did not include any specific training information, it is likely that they had access to a large GPU cluster for training. We had access to two GPUs with a combined RAM of 20GB, which proved to be insufficient for storing the parameters of the full WaveNet model. Using a smaller version of the model with fewer residual blocks, the model was able to learn some features of the raw audio. Moving forward in this research, we intend to try smaller audio samples (we trained with 30-second audio clips) and add a third GPU to train a deeper model. We also may explore funding to rent GPU time from a cloud provider.

Future work includes tuning the hyperparameters of the model, training on the full FMA dataset (1TB of raw audio), fine-tuning the model on our jingle dataset, and conditioning the model for mood-based results.

8. REFERENCES

[1] 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," arXiv preprint arXiv:1609.03499, 2016.

[2] K. Chen, W. Zhang, S. Dubnov, G. Xia and W. Li, "The Effect of Explicit Structure Encoding of Deep Neural Networks for Symbolic Music Generation," 2019 International Workshop on Multilayer Music Representation and Processing (MMRP), Milan, Italy, 2019, pp. 77-84, doi: 10.1109/MMRP.2019.00022.

[3] R. Manzelli, et al., "An end to end model for automatic music generation: Combining deep raw and symbolic audio networks," in Proceedings of the Musical Metacreation Workshop at the 9th International Conference on Computational Creativity, Salamanca, Spain, 2018.

[4] S. Luo, "Bach Genre Music Generation with WaveNet—A Steerable CNN-based Method with Different Temperature Parameters," in Proceedings of the 4th International Conference on Intelligent Science and Technology, 2022, pp. 40-46.

[5] "BandNet: A Neural Network-based, Multi-Instrument Beatles-Style MIDI Music Composition Machine," arXiv preprint arXiv:1812.07126, 2018

[6] T. Le Paine *et al.*, "Fast Wavenet Generation Algorithm," Nov. 2016.

[7] C. Donahue, H. H. Mao, Y. E. Li, G. W. Cottrell, and J. McAuley, "LakhNES: Improving multi-instrumental music generation with cross-domain pre-training." arXiv, Jul. 10, 2019. Accessed: Nov. 17, 2023. [Online]. Available: http://arxiv.org/abs/1907.04868

[8] S. Vasquez and M. Lewis, "MelNet: A Generative Model for Audio in the Frequency Domain." arXiv, Jun. 04, 2019. Accessed: Nov. 17, 2023. [Online]. Available: http://arxiv.org/abs/1906.01083

[9] W. Ping, K. Peng, and J. Chen, "ClariNet: Parallel Wave Generation in End-to-End Text-to-Speech." arXiv, Feb. 21, 2019. Accessed: Nov. 18, 2023. [Online]. Available: http://arxiv.org/abs/1807.07281

[10] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "FMA: a Dataset for Music Analysis," Sep. 2017. Available: https://arxiv.org/pdf/1612.01840.pdf

[11] Hibou Music Library, "Jingle and Jingles," www.hibou-music.com. https://www.hibou-music.com/jingle-jingles.html (accessed Nov. 09, 2023)

[12] Pixabay, "Royalty Free Music Downloads," Pixabay. https://pixabay.com/music/ (accessed Nov. 10, 2023).

[13] Freepik, "Videvo: Royalty Free Music Download Background Stock Audio," Royalty Free Music Download Background Stock Audio, 2023. https://www.videvo.net/royalty-free-music/ (accessed Nov. 10, 2023).

[14] H. Chahidi, "Royalty Free Music | Jingles," Music Screen, 2023. https://www.musicscreen.org/royalty-free-jingle-music.php (accessed Nov. 09, 2023).

[15] Tribe of Noise, "Free Music Archive," Free Music Archive, 2023. https://freemusicarchive.org (accessed Nov. 29, 2023).