Excuse Me? I Didn't Quite Catch That?: Audio Super Resolution with Generative Adversarial Networks

Arjun Sawhney, Jonathan Gomes Selman, Woodrow Wang

May 18, 2018

1 Introduction

With the rise of personal assistant systems and auditory data, auditory inputs toward technological devices are becoming more and more prevalent; however, given the coarsity and variability of sounds and subtle differences in recording devices, systems that take audio as input often have to deal with poor quality audio and at times must re-confirm or repeatedly ask the same questions to interpret the input. As such, a network that could take poor quality audio as input and enhance, or super-resolve, it without requiring confirmation or repetition from the user could improve the experience of personal assistants and other technologies that use audio data to inform actions.

In particular, we will attempt to improve an existing model which performs "bandwidth extension" by reconstructing high-quality audio from a low-quality, down-sampled input containing around 15-50% of the original samples (Kuleshov et al., 2017). We aim to use a generative adversial network architecture to improve the network introduced by Kuleshov et al. By using a version of their network as the generator and designing our own discriminator, we seek to test the potency for using a GAN to improve a promising model. Through seeing the success of WaveGAN (Donahue et al., 2018) and SRGAN (Ledig et al., 2016), we hope the application of GANs to the audio super-resolution, or bandwidth extension, problem can produce useful results to help improve the task of increasing the temporal resolution of audio signals.

2 Dataset

The data we are using comes from the CSTR VCTK Corpus provided by the Center for Speech and Technology Research. This data set includes speech data from 109 native English speakers, although we are only training on data from a single speaker for the sake of efficiency and limited computational power. The data is in the format of WAV files, which we convert to a numpy array using Python's librosa library with a specified sampling rate. We can represent an audio signal from the WAV files as a function $f(t) : [0, T] \to \mathbb{R}$, where $f(t)$ is the amplitude at t and T is the length of the signal. To process the continuous signal as an input, we must discretize $f(t)$ into a vector $x(t) : [\frac{1}{R}, \frac{2}{R}, ..., \frac{RT}{R}]$, where R is the sampling rate of the input in Hz. Notr that due to the variance of utterance lengths in our dataset, we had to pad our audio signal vectors to a specified length of 100000 discrete units. For this audio super-resolution task, we consider R as the resolution of the input x.

One difficulty is that the generator model, takes a few days to train (on a GPU) on all the data, containing multiple speakers, in the VCTK corpus which is unfeasible given our current access to compute. As such, we focus on training the model on just one speaker, which takes a few hours to train on a GPU. Given this constraint, for our current models, we split the data randomly into approximately 80% training data and 20% validation/test data. Here is a link to a few examples of WAV audio files from our dataset:

https://drive.google.com/drive/u/0/folders/1XF76CzLrHA3NoidgCi7pNkKfakDAWt9N

Although we are training on one speaker, the VCTK dataset is a promising resource as it contains a massive variety of utterances from multiple speakers, which can provide additional data to the super

resolution model in the future and can be used to ensure the model does not overfit to a single speaker. With additional computational power, beyond what's provided by AWS for this course, we would love to explore using our GAN to improve audio super-resolution for multiple speakers.

We wrote multiple scripts to process the dataset and label generated data and ground truth data for the supervised learning task of the discriminator. We decided to write our input matrices into a CSV, which can be read into a Pandas dataframe and used appropriately for the models. As a useful visualization, we can use librosa's built in functions to convert the amplitude vectors into png files for analysis, as shown below:

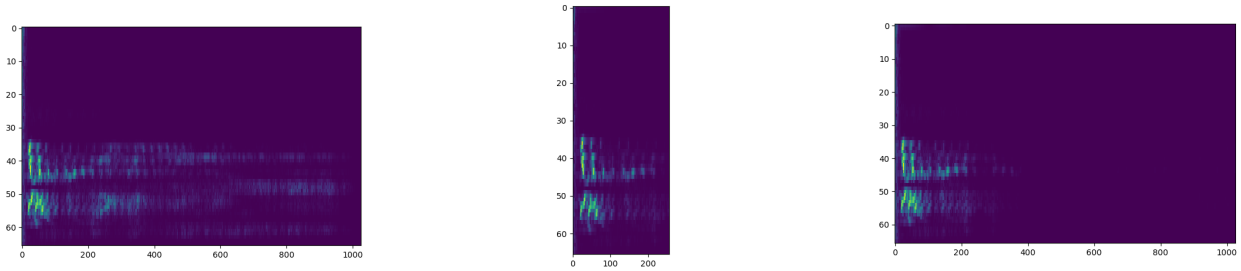

Figure 1: High resolution, low resolution, and generated image from left to right

3 Approach

Our current approach aims to create a generative adversarial network to refine the results of the deep residual network created by Kuleshov et al. Using their initial model (after re-training on the specific parts of the VCTK corpus we are using) as the generator and implementing our own discriminator, we attempt the notoriously difficult task of training a GAN.

First, we had to interpret and reformulate the existing code for the generator by cloning and adapting the required scripts from Mr. Kuleshov's GitHub repository to suit our attempted task. This took a significant amount of time, given the particular file formats and approaches utilized by Kuleshov et al. We had to write several data processing scripts in order to account for our limited access to compute and additionally account for data formatting and labelling to input to our eventual discriminator. These scripts, along with some of those cloned and reformulated from Dr. Kuleshov's repository are shown in our Github repository. The generator network involves a multitude of constituent components. Initially, the network takes in a high-resolution audio file converting it to the vector representation of audio aforementioned, which it then decimates using the Scipy python module at specified sampling rates creating "low" resolution versions. These "low" resolution vectors are then fed into a multi-layer CNN which aims to recover the original high-quality file through super resolution. We plan to extend this format to a GAN by using our initial high resolution audio vector from the data set as the ground-truth of audio data, whilst the predicted vectors recovered from the corresponding low-resolution version of the ground truth are labeled as "generated"

Note that we did not have the initial computational resources to train the generator for many epochs, so the current generator performs rather poorly, and thus makes the initial job of the discriminator even easier.

For the discriminator, we seek to create a network which can differentiate ground truth audio files from generated super-resolved audio files through binary classification. As emphasized in lecture and the paper on WaveGAN (Donahue et al.), an important aspect of training a GANs effectively is having a generator and discriminator of similar quality to avoid one network dominating the other and preventing learning. Therefore, since we want to improve the quality of a previously trained generator model we eventually desire to have a discriminator that matches the performance and complexity of our generator. We initially experimented with and implemented a basic logistic regression discriminator model. But for a slightly more robust baseline, we instead decided on a (baseline) shallow neural network for the discriminator model. We use a fully connected architecture with a single hidden layer consisting of 100 hidden units and a Relu non-linearity which is followed by a single output binary classification unit. Additionally we use an Adam Optimizer with a learning rate of 0.05 and because our initial discriminator solves a binary classification task we use the binary cross entropy loss function. As expected, since the task of the discriminator is easier

than the task of the generator, our accuracy with a shallow neural network and only 10 training epochs is 97.75%.

We have thus created the constituent parts of the GAN and have worked to combine the networks into a single model. This is the approach we hope to take moving forward.

From now until the final deadline, we will improve upon our discriminator, perhaps taking inspiration from the discriminator in WaveGAN. We plan on looking into adding convolutional layers to our discriminator and anticipate that similar features that the generator uses to perform audio super-resolution may be beneficial to the discriminator. We must be wary of common pitfalls in training GANS and in particular of falling into local optima and making sure our networks continue to improve upon each other and not be deceived by initial high accuracies. We intend to initially overlay Dr.Kuleshov's network with virtual batchnorm and adapt the cost function of the generator, to the non-saturating cost function for GANS as described in CS230 lecture. Beyond this, we may try to manipulate the architecture of the generator network itself, by adding or removing layers and experimenting with different approaches to the problem of audio super resolution.

We will use the following cost functions for the discriminator and generator respectively:

$$J^{(D)} = -\frac{1}{m_{real}} \sum_{i=1}^{m_{real}} y_{real}^{(i)} \log(D(x^{(i)})) - \frac{1}{m_{real}} \sum_{i=1}^{m_{gen}} (1 - y_{gen}^{(i)} \log(1 - D(G(z^{(i)})))$$

$$J^{(G)} = -\frac{1}{m_{gen}} \sum_{i=1}^{m_{gen}} \log(D(G(z^{(i)})))$$

4 Contributions

We feel that our group dynamic has been really solid and well balanced. During the research phase we each spent considerable time reading important literature and understanding Mr. Kuleshov's implementation. Our first steps involved each of us separately going through the process of deciphering Mr. Kuleshov's model to process the data, train the model, and then produce super-resolved audio for the single speaker data. Next, we have divided the project work into several distinct parts and worked largely collectively on each separate part. Arjun and Jonathan worked to generate a basic discriminator network to later use when training our GAN. They worked with several different basic network architectures to produce a baseline discriminator that can be further trained while training the GANs. After putting together a baseline discriminator network we all worked together to generate a data set to train our discriminator. During this process, Woody and Arjun worked collectively to understand the data processing framework provided by Mr. Kuleshov and to incorporate new scripts to generate training examples for our discriminator model. Collectively we worked to train our discriminator.

5 References

1. Mr. Kuleshov's paper on audio super-resolution using deep neural networks: https://arxiv.org/pdf/1708.00853.pdf
2. Mr. Ledig's paper on SRGAN and image super-resolution: https://arxiv.org/abs/1609.04802
3. Mr. Donahue's paper on WaveGAN, which synthesizes audio using a GAN: https://arxiv.org/abs/1802.04208
4. Link to Github: https://github.com/jonathangomesselman/CS230-Project