Audio Diffusion Model

Anonymous Author(s)
Affiliation
Address
email

1 Introduction

The goal of this project is to implement a diffusion model on audio data. I was motivated by the music segment played in class which was generated by a piece of poem. Due to the complexity of such a process, professor Dubnov advised me to start with the implementation of a diffusion model. Diffusion models are used as generative models to reconstruct ruined data by understanding the concept behind the ruined data. Therefore, diffusion models can be seen as a starting point of music generation.

2 Related Work

Diffusion models and U-Net models are generally used in audio classification in papers. To narrow the scale of my project, I focused on reconstruction of audio data. Paper Denoising Diffusion Probabilistic Models suggested a U-Net model on image dataset, where I fitted an audio dataset with a similar approach of U-Net model.

3 Architecture

3.1 Dataset

The dataset selected for this process is a dataset of various speakers pronouncing single digits. 60 speakers vary in backgrounds, accents, ages, and genders thus adding diversity to the dataset. Due to the uncontrollable nature of human voice, each audio sample differs in audio length and power. This dataset is selected for different digits creates a diversity for model training, also the regularity of pronunciation creates a clear target for audio reconstruction.

3.2 Mel-Spectrogram

Figure 1 shows a triangular filter bank with 40 filters on a Mel-scale. The y-axis of Figure 1 shows the weight multiplied to frequencies. It is shown that the center points of each frequency band are multiplied with 1.0, meaning that the entire frequency is preserved. Spreading out from the center point of each filter, the weights are gradually decreasing to weaken corresponding frequencies. Moreover, the filters are more spread out as the frequency increases, resulting in less emphasis on high frequencies.

Mel-spectrogram is selected due to its nature of emphasizing lower frequencies and putting less emphasis on high frequencies. Such a nature mimics the non-linear sound the human ear perceives, and can better represent our dataset consisting of human voice. Figure 2 shows a comparison of applying mel-scale spectrogram versus linear spectrogram. It is clearly seen that linear spectrograms compress information in a lower frequency range, thus limiting the display of information. On the other hand, mel-scale spectrograms stretch the lower frequencies and create a better presentation of human voice audio samples.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

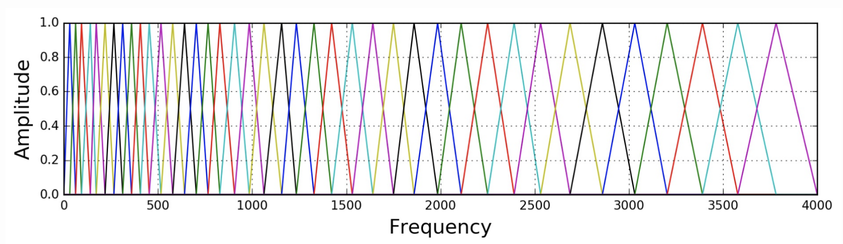


Figure 1: 40 filters filter bank on a mel-scale

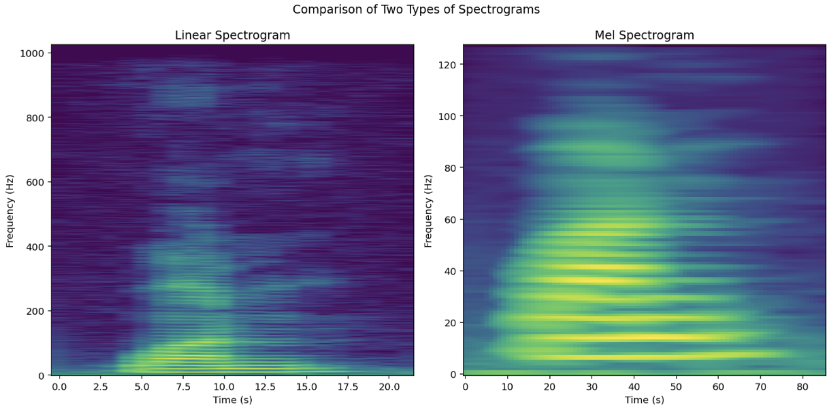


Figure 2: Comparison of mel-scale spectrogram and linear spectrogram

3.3 Data Augmentation: Forward Process

After we obtain the processed data matrix from mel-spectrogram, we apply data augmentation to each sample. Data augmentation adds layers of “noise” to samples by multiplying weights to the mel-spectrogram matrix. This process aims to create an easy access dictionary to each layer of augmentation. Image distribution can be described using variance β, each step increases the value β to reach a slightly more corrupted image following the formula, where t represents the layer number and I is a fixed variance value.

$$\begin{aligned} q(x_t|x_{t-1}) &= N(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \\ &= \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon \end{aligned}$$

3.4 U-Net Model

We apply dataset consisting of pre-processed mel-scale spectrogram matrices into a U-net model, which trains the model to predict “useless” voice and learn to remove then reconstruct the original spectrogram. The U-Net model we use consists of 4 layers of encoder and decoders, expanding the image channel from 3 to 1024 and back to 1. In each step, the loss function is calculated by taking the difference of predicted voice and the actual added noise which the forward process directly calculates.

The U-net Model is selected for its nature of doing classification locally on every pixel, the method is suitable for small scale image evaluation like the dataset we selected. Also for its ability to handle high-resolution images and thus reconstruct accurate segmentation.

4 Experiment

Dataset status The dataset consists of 30,000 audio samples where 60 speakers repeat each digit, from 0 to 9, 50 times. We randomly selected 6000 samples for the train dataset and 2400 samples for the test dataset.

Figure 3 shows the scattered distribution of audio length in the training dataset. Thus compares three consecutive records from a speaker recording the same digit, and demonstrates that even consecutive

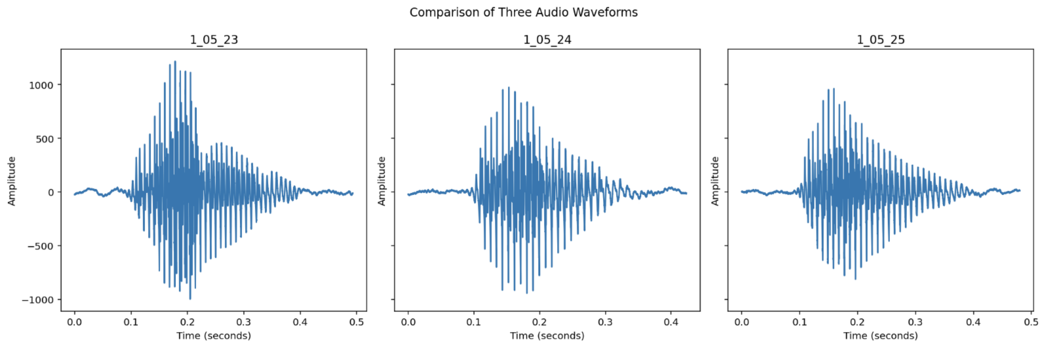


Figure 3: display of three audio samples from one speaker

recording generates various sound amplitude and audio length. Data processing targets such variations between audio samples.

Mel-spectrogram Data is imported to working space using librosa.load function, which transforms a WAV file to an array of complex numbers. Then, mel-spectrogram transformation is performed on the array of complex numbers. The process begins with passing a signal through a pre-emphasis filter, then the signal is divided into overlapping frames and applying each frame with a window function. Next, Fourier transform, in our case Short-Time Fourier Transform, is performed individually on each frame to calculate the power spectrum. Then, amplitude is converted to DB. Finally, a mel-filter bank is computed and applied on the computed matrix.

Mathematically, we can convert between Hertz (f) and Mel (m) using the following equation:

$$m = 2595 log_{10}(1 + \frac{f}{700})$$

$$f = 700(10^{m/2595} - 1)$$

And model the filter bank using the following equation:

$$H_m(k) = \begin{cases} 0 & k < f(m) \\ \frac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \leq k \leq f(m) \\ \frac{f(m+1) - k}{f(m+1) - f(m)} & f(m) \leq k \leq f(m+1) \\ 0 & k > f(m+1) \end{cases} \tag{1}$$

Data Augmentation To implement the Forward process, we set 300 beta values to be equally ranged from [0.0001, 0.02], meaning getting a corrupted image from 0 to 0.02 takes 300 steps. Then, we altered the variance β to $1 - \alpha$, making α represent the percentage of the previous image in a new image. To improve machine runtime, we pre-calculated α for each step so that each step can be calculated directly from the original image with the following equation:

$$q(x_0|x_{t-1}) = \sqrt{a_t a_{t-1} ... a_1 a_0} x_0 + \sqrt{1 - a_t a_{t-1} ... a_1 a_0} \epsilon$$

Data pre-process We are going to fit the matrices of Mel-spectrograms into the U-Net model. Before training the matrices, we need to pad the matrices into suitable sizes for model training. First, the length of data audios are not the same, so we took the maximum length of audio in training data and forced such length on other audio samples to unify the length of sample to [128,44]. Then, we realized the U-Net model requires data of equal height and length, where the height and length should be power of 2, to perform dimensional editions. So we changed the mel-bin of the spectrogram from 128 to 64, and performed zero padding where pad = 20 to data, and thus reached consistency data shape [64,64].

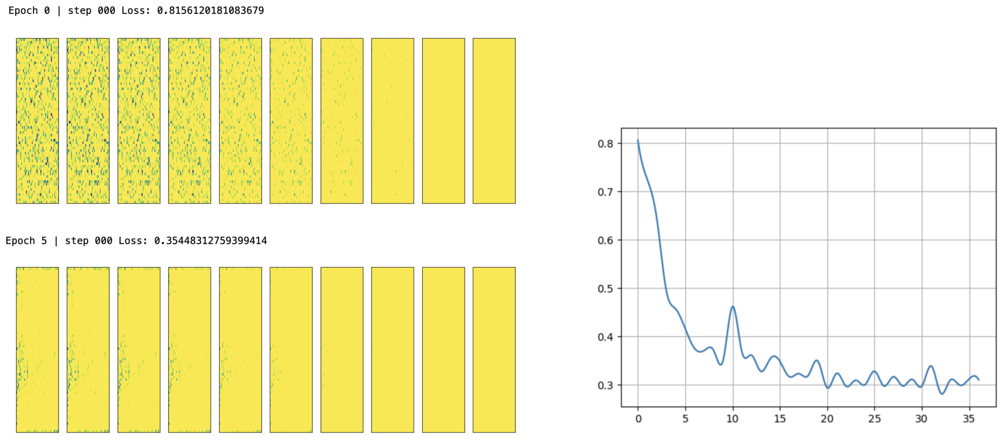
Convolutions Layer We used a padded convolutional layer instead of an unpadded convolutional layer as suggested in the paper. The first filter applied is a 2d convolutional layer which is used to extract features from input, thus enlarge the channel of input sample and creates a feature map. Following a batch normalization is applied to normalize the distribution of intermediate layers. Lastly, we pass each value of the feature map through an rectified linear unit(ReLu) as an activation function.

Encoder Encoder layers are the contraction process of the U-Net model, aiming to gradually lower the spatial dimension of the input sample. Encoder block consists of a convolutional block and a pooling layer. We implemented a max pooling filter which extracts the local maximum of a feature map to reduce the spatial dimension. Encoder block is repeated 4 times, until feature map channels are increased to 1024 and hits the bottle neck.

Decoder Decoder layers extract the features from the feature map and perform reconstruction. Decoder is implemented with an up convolutional step which extracts features from feature maps and makes predictions on how to reconstruct a feature map that doubles in spatial dimension. Following, we apply the convolutional layer again to decrease channel size.

Skip connection The process of up-scaling and down-scaling loses information through the build and extraction of feature maps. Therefore we included a skip connection that reintroduces details from the encoder directly to the corresponding decoder layer.

5 Results



(a) Display of Epoch 0 and 5

(b) Loss value of Epoches 0-40

Figure 4

To train the model, I runned 40 epoches on 6000 samples with batch size 128. As shown in Figure 4, we reached a clear decrease in loss value in model training, and spectrogram visualization shows a clear improvement in signal reconstruction through the process of training. However, timing was a big issue in my model training. We maintained a lower resolution of input samples by controlling the input size to 64 by 64 pixels and enlarged mel-scale spectrogram bandwidth to 1024. One possibility of such time consumption is the step value 300 selected in forward process training, which could be unnecessary due to the small size of input samples.

6 Conclusion

In this project, I fitted audio samples in a diffusion model based on an original model that takes in image samples. In order to mimic an image sample, audio samples are required to go through a spectrogram transformation. However, information is lost in such transforms and I selected a mel-scale spectrogram to preserve as much useful information as needed. Moreover, in terms of fitting the U-Net model, it is crucial to understand the underlying meaning of required input shape. Since audio samples do not come in consistent size nor range, data pre-processing is very important in this project.

7 Link to Github Repository

https://github.com/astoriama/CSE190_final_project

References

[1] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising Diffusion Probabilistic Models. https://doi.org/10.48550/ARXIV.2006.11239

[2] Mel-spectrogram: Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between. (n.d.).

[3] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In arXiv [cs.CV]. http://arxiv.org/abs/1505.04597