Final Project Report 1

Armani Rodriguez

Background

Speech synthesis is a problem that has been iterated on for hundreds of years, with speech synthesizers often relying on prerecorded phonetics and combining them to create comprehensible, albeit choppy recreations of speech. Machine learning approaches to generating audio have been met with mixed success depending on the type of audio data, with the time series aspect of the data adding additional complexity. Automated high quality humanlike speech synthesis is very sought mainly due to its myriad of applications in the accessibility industry, where speech synthesis aids those with visual or auditory impairments.

This project aims to explore the feasibility of using an unsupervised generative model known as the Vector Quantization Variational Autoencoder (VQ-VAE) for speech synthesis[1]. This model is the backbone of many revolutionary generative computer vision models including OpenAI's DALL-E. By leveraging this model's ability to generate reconstructions of training data from a lower dimensional latent space along with the sequential prediction power of the transformer in order to synthesize speech, I hope to craft a novel approach to speech synthesis[2].

I will judge the quality of the generated speech by both my own perception, and more formally with a deep classifier trained on the original data.

Data

This project uses the AudioMNIST dataset, comprising of 30,000 recordings of spoken digits (0-9) spoken by 60 different speakers of both genders and various ethnicities in the WAV audio format.

Because our model is not one built to handle time series data such as audio recordings, the data must be preprocessed into a usable format. My data preprocessing pipeline is as follows:

1. Each recording is resampled to a uniform sample rate of 22,050 hertz.

2. The leading and trailing silence on either end of each recording is trimmed off.

3. The size of each recording is fixed such that each contains 22,050 samples.

4. Each recording is transformed into its Mel spectrogram representation. The spectrogram representation visualizes the spectrum of frequencies in a waveform as it varies over time, making it a good candidate for our model. The Mel spectrogram representation uses the Mel frequency scale, which mimics human perception by being more sensitive to frequency changes at lower frequencies. Because our data comprises of human speech and human perception of synthesized speech is important, I used the Mel spectrogram representation.

5. The magnitude of the frequencies in the Mel spectrogram are expressed in linear amplitude units, which do not correspond well to human perception of sound. Humans perceive sound on the decibel (dB) scale, therefore the next step in my preprocessing pipeline is to convert the amplitude units to decibels.

6. We finally normalize the data using min-max normalization.

Of course, humans cannot audibly perceive spectrograms, thus we perform the inverse of these transformations on the reconstructions.

A Simple Autoencoder

An autoencoder is an unsupervised learning model that compresses it's input to a lower dimensional latent space and then attempting to reconstruct it from that lower dimensional space. My preliminary model is a convolutional autoencoder consisting of only two components: The encoder denoted E and the decoder denoted D. The process of reconstructing an unlabelled input x is simply $\hat{x} = D(E(x))$. For the loss I use only one term, the reconstruction loss, which is simply the mean square error between x and $\hat{x}$.

This model is not a generative model, but it plays a major role in this project and the development of the VQ-VAE model. This model serves as a baseline for the results, a vessel for quicker hyper parameter tuning, and enabling transfer learning to the more complex generative model.

VQ-VAE

The step from the described simple autoencoder to the massively popular generative model known as VQ-VAE is not a big of a jump as one might imagine. The VQ-VAE contains the same two components as the preliminary auto encoder, with just an added component known as the vector-quantization codebook. This codebook denoted Q consists of N "codewords" or discrete vectors of dimension D. The process of vector quantization is mapping an input of dimension D to it's nearest neighbor in the codebook. These codewords are trainable parameters which attempt to minimize the distance between each input and it's nearest neighbor, thus in theory attempting to approximate the identity function. This vector quantization step sits between the encoder and decoder. This model has two additional terms in the loss function in addition to the reconstruction loss which attempt to move the output of the encoder closer to the codewords while simultaneously moving the codewords closer to the output of the encoder. The complete loss function is as follows:

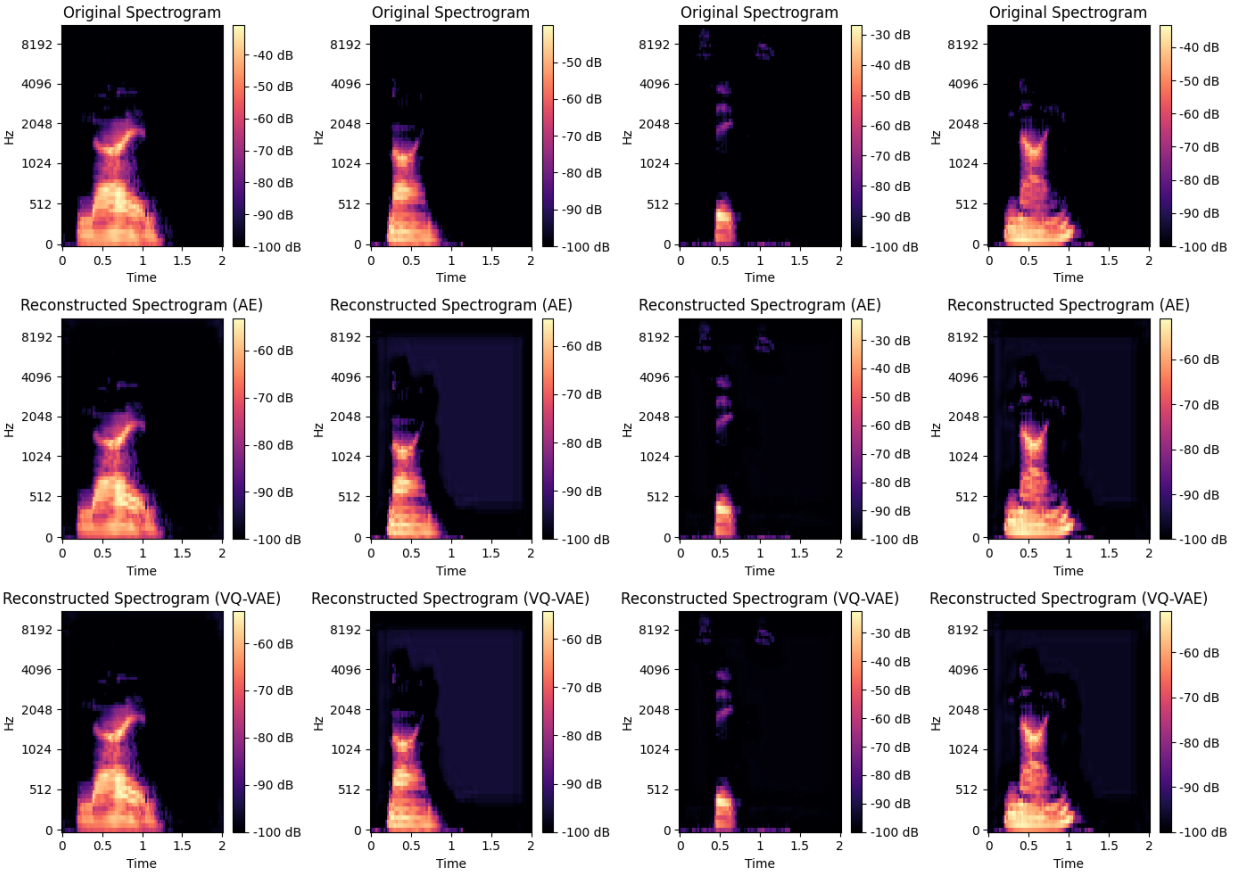
$$\mathscr{L}(x, \hat{x}) = \text{MSE}(x, \hat{x}) + ||\,\text{sg}[E(x)] - q\,||^2 + \beta\,||\,E(x) - \text{sg}[q]\,||^2$$

Where q is the chosen codeword $Q(E(x))$, β is a weighing hyperparameter, and sg denotes the stop gradient function. The generative property of this model comes from the fact that by choosing codewords and passing them through the trained decoder, one can generate new data[1].

Preliminary Results

Dataset	Classification Accuracy (Validation Set)
Original Dataset	0.9946
Autoencoder (64 Epochs)	0.995
VQ-VAE (64 Epochs, 32 Codebook Dimension, 256 Codewords, Beta=0.25)	0.833

Reconstructions from both the auto encoder and VQ-VAE generated perceivable audio that matched their labels, with the autoencoder audio being almost indiscernible from the originals while the VQ-VAE audio while perceivable is audibly distorted.



The figure above displays the mel spectrogram representation of original data, along with the auto encoder and VQ-VAE reconstructed versions of said spectrograms.

Next Steps

While I continue to fine tune the VQ-VAE, a process made difficult by the long iteration time of training this model, I will move forward with the final component of this project: the transformer. I have decided that because MiniGPT is an implementation man have found success with in implementation of VQ-GAN, a generative image model utilizing the same architecture but for image data [2], I will also use MiniGPT. Coming in at a not-so mini 13 billion parameters, MiniGPT will take quite the effort to train. The good news that by utilizing transfer learning, I can train MiniGPT while also fine tuning the VQ-VAE model.

Citations

1. Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2018.

2. Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis, 2021