Encoding Audio with DFT

Anthony Shara, Adi Ganapathi, Dohyun Cheon, Larry Yan, Richard Shuai

December 11, 2020

1 Encoding Audio Data to Explore Hidden Features

Today, many technologies such as Amazon Alexa and Siri rely on machine learning to detect audio patterns in there surrounding environments. Like any machine learning task, the objective when learning about audio signal is to learn and make predictions about about data - in this case sound.

If we think about human speech, different words are characterized by different patterns in the sound of our voices. When collecting data from a microphone, the data represents the varying strength of the audio signal over the time duration of the sample. The question now is what features of this audio signal allow us to best learn about to recognize patterns in sound?

2 Wave Fundamentals

In order to understand features of sound waves from an audio signal, we must first review the basics of waves.

Every fundamental periodic wave consists of an amplitude and a wavelength (See Figure 1). The amplitude (A) of the wave is the distance from the center of the wave to the peak of the wave. The wavelength (λ) is the distance between two equivalent points on the wave. Since 1 rotation around the unit circle is 2π, the frequency (f) is expressed as $\frac{2\pi}{\lambda}$. Notice the the frequency (f) is the number of times that we complete one rotation of the unit circle per wavelength. Every wave takes the general form of $f(x) = A\cos(\frac{2\pi}{\lambda} \cdot x) = A\cos(f \cdot x)$.

Now lets look at a more complex wave (as shown in Figure 2). Although this wave look complex at first sight, it simply breaks down into $\cos(5x)\ + \cos(2x)$. Notice how each contributing wave has its own amplitude and frequency. Since each contributing wave is characterized by a distinct amplitude and frequency, you may be asking if we can use this information to unpack information in an audio signal that is sitting in plain sight!

Now, we know that a complex wave can be represented as a linear combination of fundamental waves where each fundamental wave is characterized by a unique amplitude and wavelength. This tells us that we can learn about any signal by

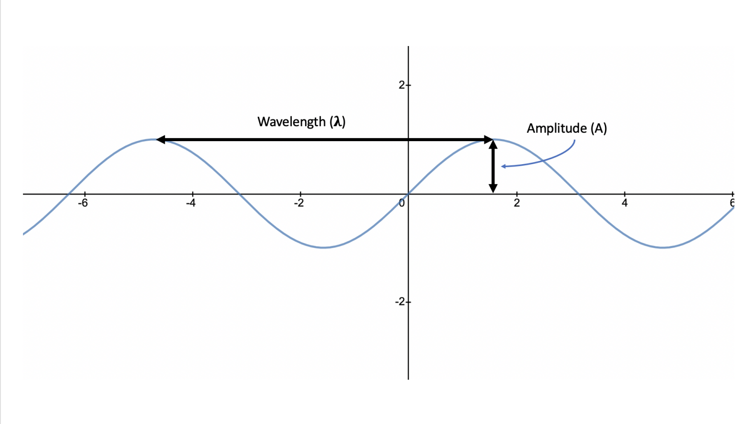

Figure 1: sine wave labeled with amplitude and wavelength

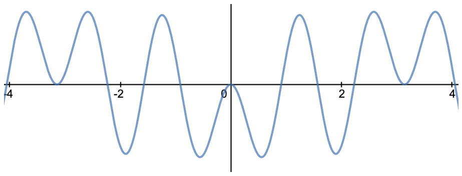

Figure 2: plot of $\cos{(5x)}\ +\cos{(2x)}$

analyzing how its amplitude over time, the strength of its frequencies, or some combination of the two!

3 Sampling

Before we begin to discuss characteristics of audio signals, lets take a quick detour and talk about sampling. Digital signal processing is the bridge between the continuous time world that we live in and the discrete time world of computers. In order for us to work with audio signals, digital signal processing is necessary every step of the way! In order to turn a continuous signal into discrete signals process-able by a computer, we rely on the **Nyquist–Shannon sampling theorem** which states that a signal can be sampled and perfectly reconstructed from its samples if the waveform is sampled over twice as fast as it's highest frequency component ($f_s > 2 \cdot f_{max}$). This means that if we sample fast enough, we can turn perfectly turn a continuous signal into a discrete signal! In figures 4 and 5 we see that the quicker the sampling frequency, the closer you get to the actual signal.

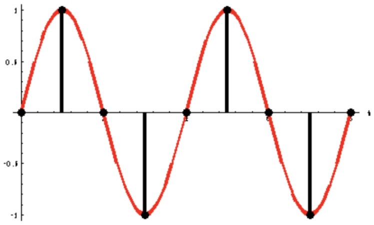

Figure 3: slower sampling rate (source: Berkeley microscopy)

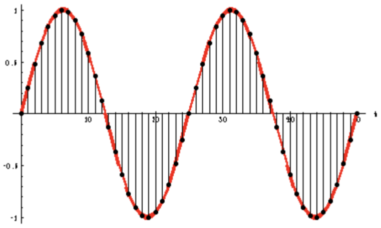

Figure 4: faster sampling rate (source: Berkeley microscopy)

4 Analyzing a Signal in the Frequency Domain

When learning about patterns in audio signals, the most obvious approach is to look at the change in amplitude over time. However, with this approach, it is difficult to learn about the different frequencies that live within the signal meaning that you are loosing important information that characterizes your data! This leads to the conclusion that it cannot be a very good approach.

4.1 The Discrete Fourier Transform (DFT)

Since analyzing the amplitude of the wave at different points in time is not a good approach, you might be wondering how else you can analyze your signal. As mentioned earlier, waves can be characterized by their amplitude and frequency. This leads you to believe that the best way for your data to encapsulate all of the information that describes each contributing wave. This is the role of the **Fourier Transform**. The Fourier Transform is a function that decomposes a signal into its constituent frequencies!

Since an audio signal is constructed via sampling and therefore a discrete signal, we will use the Discrete Fourier Transform (DFT). The Fourier Transform of a discrete time, aperiodic signal is defined as:

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-i\omega n}$$

were F is a function of the frequency, ω.

This means that the Fourier transform is simply a linear combination of complex exponentials corresponding to each constituent frequency in the signal weighted by the energy of the frequencies from each contributing wave. Recall that $e^{-i\omega n} = \cos(\omega) + i\sin(\omega)$. Since the DFT give you a complex signal, we are mostly interested in the magnitude of the Fourier Transform. The **Magnitude Response** tells you about the magnitudes (energy) of each frequency contained in the signal. Figures shows plots of the raw signal and magnitude response of a person saying the word "zero".

This tool is the key to unlocking all of the data contained within an audio signal!

4.2 Sound Waves

Now that we know about wave basics and Fourier Transforms, lets explore how they relate to sound waves! The most simple form of audio is a single tone such as that from a tone generator or a dog whistle. These kinds of sounds contain only a single frequency. The average human can only hear frequencies between 20 Hz and 20 kHz. This range is known as the audible spectrum. Figures 6 and 7 show the raw signal and magnitude response of a 20Hz and 200 kHz tone.

Notice that each tone has a spike at $\omega = f/2$ where f is the frequency of the tone and the each tone is represented by shifted cosine wave. In general $\cos(\omega_0 x)$ Fourier Transform of a similar form. It has a spike at $\omega = \pm\frac{\omega_0}{2}$. Further, a phase change in time corresponds to a shift in frequency. Also, a single frequency at $\omega = \alpha$ corresponds to a spike only at $\alpha on the \omega axis$. Mathematically, a phase shift of a single frequency in time is denoted as $e^{i\omega}$ and each of these spikes is

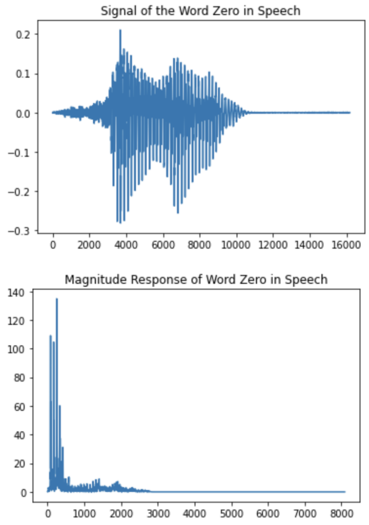

Figure 5: raw signal(top) and magnitude response (bottom) of the word "zero" in speech

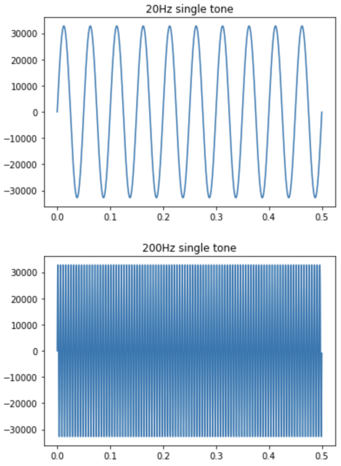

Figure 6: 20 Hz tone (top) and 200 Hz tone (bottom)

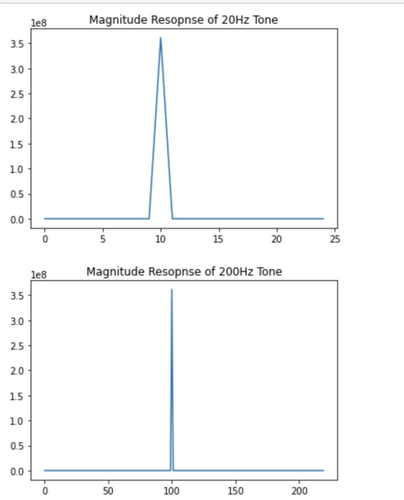

Figure 7: magnitude response of 20 Hz tone (top) and magnitude response of 200 Hz tone (bottom)

represented by the delta function which simply put represents a single point on the graph and is defined as having infinite height and no width. The height that we draw is simply the strength of the delta function.

These are the basic building blocks for interpreting the DFT of a wave, however, in the case of an audio signal, these waves are always changing and learning about the entire signal as a whole is not always sufficient. Since the signal is dynamic,we need to be able to capture the changing story of this signal over time. Time and frequency descriptions alone are no longer sufficient and we need a way to tie them together.

5 Short Term Fourier Transform (STFT)

Recall the sampling theorem which tells us that by taking samples of a continuous signal we can perfectly reconstruct it as a discrete signal. Let's use this as motivation to bring together the time and frequency descriptions of our signal and once and for all tell the true story of our signal. In the same way that we would sample a audio to create a digital signal, let's break our audio signal into small windows where we can take a snapshot of what is going on at that point in time. By taking the DFT of each of these snapshots throughout the entire length of the audio signal, we are able to learn about the the strength of each contributing frequencies the signal at each point int time! Each Fourier Transform gives you the Fourier Spectrum at a specific time. This is critical in capturing the details hidden inside of an audio signal.

This process is known as the **Short Term Fourier Transform (STFT)**. The STFT creates an image which shows the change in Fourier Spectrum over time by sliding a window funtion over discrete intervals of the time varying signal and taking the DFT of each window. This action should sound familiar to you. This action translates to the convolution of a window function with a time varying signal and then taking the DFT! Mathematically, this can be expressed as:

$$X[n, \omega) = \sum_{m=-\infty}^{\infty} x[n+m]w[m]e^{-i\omega m}$$

Here, x is a time varying signal and w is a window function (such as, but not limitted to the rectangle function). It is important to notice the 'm' in the complex exponential. Since we have each index in our summation (which is performing the convolution) in our complex exponential, we are taking the DFT at each point in the convolution instead of the DFT over our entire signal. Also notice that the STFT of a signal $x[n]$ is $X[n, \omega)$ which is a 2D function of n (discrete variable) and ω (continuous variable). This means that the STFT is a mapping from 1D to 2D space. This 2D mapping created by STFT is called a **Spectrograms** and is a useful tool that gives us temporal information to tell the full story of our signal! Figure 8 shows a spectrogram of a person saying the word "zero".

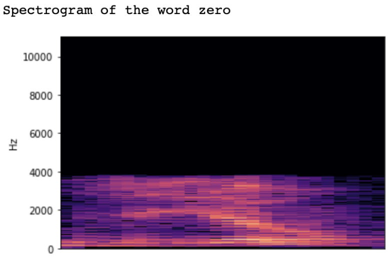

Figure 8: spectrogram of the word "zero"

6 Convolutions with the DFT

We have seen how converting signals to the frequency domain can make them easier to interpret and analyze. As a brief connection to a previous week, we want to mention that sometimes, converting images to the frequency domain using the DFT can be helpful for applying convolution.

According to the **convolution theorem**, the pointwise multiplication of two signals in the frequency domain is actually equivalent to a convolution in the time domain. Note that "convolution" in machine learning refers to what is actually a cross correlation. With this in mind, we can actually use a 2-dimensional DFT to convert an *image* into the frequency domain. By transforming both the input image and the flipped kernel to the frequency domain, we can perform a cross correlation between the image and the kernel with a pointwise multiplication. We can then use invert the DFT to transform the image back into the time domain. In this way, we can achieve what is known as an "FFT convolution." Although FFT convolutions are not commonly implemented in practice, they can be more efficient when the size of the kernel is large relative to the input image.

7 Conclusion

After exploring the sampling, waves, and the DFT, we were able to bring aspects of each of these key signal processing tools to unpack all of the data that we need to characterise and learn about patterns in audio signals. As we dove deeper, we saw that an audio signal is a complex and dynamic monster with many layers of information to unpack in order to really generate the full picture. By just looking at the signal in the time domain, we only learn about the strength of the signal at a given time. Looking at the Fourier Transform of the signal us about the energy of different frequencies living in our audio signal, this still treating our signal as a static wave. In the end, we learned that it is necessary to learn about the energy of different frequencies contributing the signal over time.

References

[1] Allen V. Oppenheim, Signals and Systems, Second Edition, 1997

[2] Berkeley Microscopy, Capturing images
http://microscopy.berkeley.edu/courses/dib/sections/02Images/sampling.html

[3] Dima Shulga, Speech Classification Using Neural Networks: The Basics
https://towardsdatascience.com/speech-classification-using-neural-networks-the-basics-e5b08d6928b7

[4] Jarno Seppänen, Audio Signal Processing basics, 1999
https://www.cs.tut.fi/sgn/arg/intro/basics.html

[5] M. Lustig, EE123 Digital Signal Processing Lecture 5B Time-Frequency Tiling, EECS UC Berkeley