In-class Assignment # 1

Due Date: Thursday, August 29, 2019 by 11:59PM

Total Points: 50

The purpose of this assignment is to gain familiarity with the audio sampling and quantization in the Python environment. The questions below are directly related to today's lecture, so look at your notes and slides for assistants. Feel free to perform internet searches for documentation on certain functions. You can also create separate functions as needed. This is an individual assignment, so please complete it on your own. However, feel free to ask questions to the instructor or your neighbor. **Upload your final code and wav files to Canvas. Detailed answers to the questions can be written as comments in your code. Be sure to indicate the question that you are answering in your comments.**

All assignments must be submitted on time to receive credit. No late work will be accepted, unless you have a prior arrangement with the instructor.

In a terminal, be sure to activate your virtual environment (see hw0 for details). Also, be sure to install the following modules: *sounddevice, soundfile, scipy.io.wavefile, librosa, numpy.*

Question 1. [25 POINTS]

Sound Recording and Quantization: This problem provides practical experience with using your computer's microphone to capture audio in a Python environment. You'll also learn how to save this recording to a file and change the bit rate (e.g. quantization).

1. Record yourself saying the following sentence: "Hi my name is [first name] and deep learning for speech processing is my favorite course." This can be accomplished with the *rec*() function that is from *sounddevice*. Use a sampling rate of 44100 Hz, 1 channel, and specify the number of frames (e.g. samples) so that the recording is around 7 seconds long. Store this recording in a variable named: *myrecording*

2. Use *soundfile*'s *write* function, to output this recording to a 'WAV' file. Generate three wav files, where one is saved in 'PCM_32' format, one is saved as 'PCM_16', and the last one is saved as 'PCM_U8'.

3. What differences in signal quality do you observe as the data format is changed? Why does this happen? What version sounds best? What version sounds worse?

Question 2. [25 POINTS]

Signal Resampling: This problem provides practical experience with using Python to change the sampling rate of an audio signal.

1. Use *librosa*'s *resample* function to change the sampling rate of the signal that was recorded. Change the sampling rate to the following, and save each output to a 'WAV' file ('PCM_32' format): 22050, 11025, 5512, 2250.

2. What differences in signal quality do you observe as the sampling rate is changed? Why does this happen? What version sounds best? What version sounds worse?