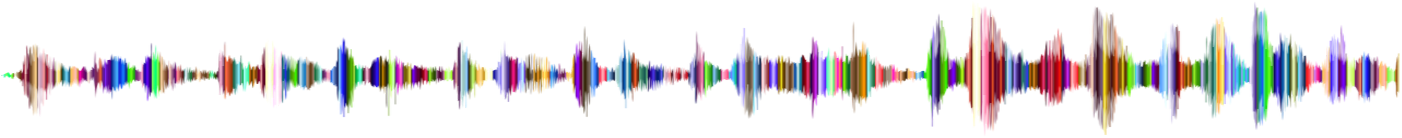
Trigger Word Detection

Welcome to the final programming assignment of this specialization!

In this week's videos, you learned about applying deep learning to speech recognition. In this assignment, you will construct a speech dataset and implement an algorithm for trigger word detection (sometimes also called keyword detection, or wake word detection).

- Trigger word detection is the technology that allows devices like Amazon Alexa, Google Home, Apple Siri, and Baidu DuerOS to wake up upon hearing a certain word.
- For this exercise, our trigger word will be "Activate." Every time it hears you say "activate," it will make a "chiming" sound.
- By the end of this assignment, you will be able to record a clip of yourself talking, and have the algorithm trigger a chime when it detects you saying "activate."
- After completing this assignment, perhaps you can also extend it to run on your laptop so that every time you say "activate" it starts up your favorite app, or turns on a network connected lamp in your house, or triggers some other event?



In this assignment you will learn to:

- Structure a speech recognition project
- Synthesize and process audio recordings to create train/dev datasets
- Train a trigger word detection model and make predictions

Updates

If you were working on the notebook before this update...

- The current notebook is version "1a".
- You can find your original work saved in the notebook with the previous version name ("v1")
- To view the file directory, go to the menu "File->Open", and this will open a new tab that shows the file directory.

List of updates

- 2.1: build the model
  - Added sample code to show how to use the Keras layers.
  - Lets student to implement the `TimeDistributed` code.
- Spelling, grammar and wording corrections.

Let's get started! Run the following cell to load the package you are going to use.

In [1]:

```
import numpy as np
from pydub import AudioSegment
import random
import sys
import io
import os
import glob
import IPython
from td_utils import *
%matplotlib inline
```

1 - Data synthesis: Creating a speech dataset

Let's start by building a dataset for your trigger word detection algorithm.

- A speech dataset should ideally be as close as possible to the application you will want to run it on.
- In this case, you'd like to detect the word "activate" in working environments (library, home, offices, open-spaces ...).
- Therefore, you need to create recordings with a mix of positive words ("activate") and negative words (random words other than activate) on different background sounds. Let's see how you can create such a dataset.

1.1 - Listening to the data

- One of your friends is helping you out on this project, and they've gone to libraries, cafes, restaurants, homes and offices all around the region to record background noises, as well as snippets of audio of people saying positive/negative words. This dataset includes people speaking in a variety of accents.
- In the raw_data directory, you can find a subset of the raw audio files of the positive words, negative words, and background noise. You will use these audio files to synthesize a dataset to train the model.
    - The "activate" directory contains positive examples of people saying the word "activate".
    - The "negatives" directory contains negative examples of people saying random words other than "activate".
    - There is one word per audio recording.
    - The "backgrounds" directory contains 10 second clips of background noise in different environments.

Run the cells below to listen to some examples.

In [2]:

```
IPython.display.Audio("./raw_data/activates/1.wav")
```

Out[2]:

0:00 / 0:00

In [3]:

```
IPython.display.Audio("./raw_data/negatives/4.wav")
```

Out[3]:

0:00 / 0:00

In [4]:

```
IPython.display.Audio("./raw_data/backgrounds/1.wav")
```

Out[4]:

0:00 / 0:00