Librosa, Espectrogramas, y las Transformadas de Fourier

**Breve introducción**

Los audios, físicamente, están compuestos por señales. De esta forma, podemos aplicar técnicas de análisis de señales sobre ellos.

Un espectrograma es un gráfico que muestra el cambio de las frecuencias de una señal en función del tiempo.

Librosa es el package (biblioteca, library) de Python que utilizamos para realizar los espectrogramas de los audios de los pacientes/voluntarios. Para esto, Librosa realiza una Transformada de Fourier de Tiempo Reducido (TFTR, STFT en inglés).

**Transformadas de Fourier**

La Transformada de Fourier (y sus variantes, como TFTR/STFT) no es más que una transformación matemática que descompone una función (en este caso señales, que son en función del tiempo) en las frecuencias que las componen.

La Transformada de Fourier es básicamente el espectro de frecuencias de una función: nos dice qué frecuencias están presentes.

La STFT utiliza una "ventana de tiempo", según Wikipedia:

*Las ventanas son funciones matemáticas usadas con frecuencia en el análisis y el procesamiento de señales para evitar las discontinuidades al principio y al final de los bloques analizados.*

*En el procesamiento de señales, una ventana se utiliza cuando el análisis se centra en una señal de longitud voluntariamente limitada. En efecto, una señal real tiene que ser de tiempo finito; además, un cálculo sólo es posible a partir de un número finito de puntos.*

*Para observar una señal en un tiempo finito, se multiplica por una función ventana.*

Las ventanas de tiempo son muy importantes e influyen en el resultado del espectrograma, debido a que uno de los problemas que tiene STFT es que, dependiendo del tamaño de la ventana, puede perder resolución de frecuencia, o resolución de tiempo.

La "resolución de frecuencia" es lo que nos permite identificar con claridad el valor de la frecuencia en el espectrograma. Si perdemos resolución de frecuencia en un espectrograma, nos va a costar distinguir el valor de la frecuencia.

La "resolución de tiempo" nos indica el tiempo en que las frecuencias cambian. Si perdemos resolución de frecuencia, en el espectrograma no vamos a notar claramente cuándo cambian esas frecuencias.

La regla es la siguiente: si disminuimos el tamaño de la ventana, aumenta la resolución de tiempo. Si aumentamos el tamaño de la ventana, aumenta la resolución de la frecuencia.

**¿Qué es n\_ftt y hop\_length?**

Uno de los parámetros de la señal es el sampling rate, la cantidad de muestras por segundos que se toman de una señal continua para convertirla en una señal discreta. n\_ftt indica qué tantas muestras por segundo debemos tomar en un segmento, mientras que hop\_length indica cuántas muestras debemos “saltear” entre segmento y segmento.

Es decir, con hop\_length y n\_ftt, dividimos a la señal en una serie de segmentos, y en base a eso calculamos la STFT.

Usualmente estos segmentos se superponen (con el fin de evitar la pérdida de información), por lo que la distancia entre dos segmentos (el hop\_length) usualmente no es n\_ftt, sino algo como n\_ftt/2.

Aparentemente, los segmentos en los que divide n\_ftt a la señal son llamados también “frames” de audio.

Cuando se computa una STFT, se computa la FFT (Fast Fourier Transform, un algoritmo eficiente para calcular la Transformada de Fourier Discreta) para un corto número de segmentos. Estos segmentos tienen como longitud el parámetro n\_fft.

**¿Qué Window Size/Length utilizar?**

Generalmente la longitud de ventana que se recomienda utilizar para las tareas de reconocimiento del habla (speech recognition), o tareas que involucren el habla en sí, ronda entre los 20-40ms.

Muchos dicen que un valor como 25ms es lo adecuado. El paper en el que nos basamos utilizó una ventana de 23ms.

**Window Size a Segundos**

Donde SR es el sampling rate del audio, N\_FFT es el término que describimos anteriormente, y T es el resultado en segundos.