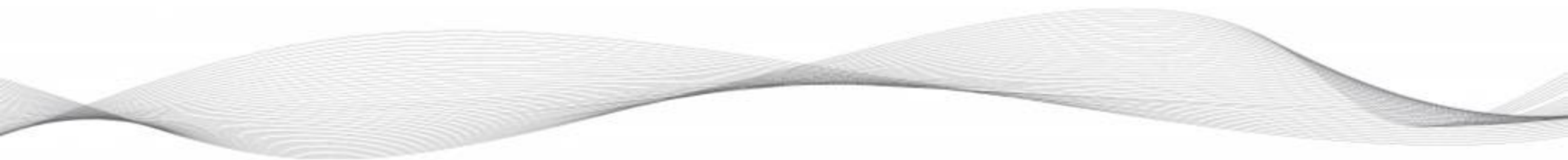


# Deep Learning para sonidos

Jose Omar Giraldo Valencia



# ¿Quién soy?



- Soy ingeniero de sonido y músico que disfruta de programar
- Toco percusión
- Actualmente trabajo en detección y clasificación de sonidos ambientales y naturales



**AAC CENTRO DE ACÚSTICA APLICADA**  
Ingeniería + Laboratorio

# Sobre esta charla ...



- Breve resumen del taller “Deep learning for MIR”, extendido a sonidos en general.
- Center for Computer Research in Music and Acoustics
- <https://ccrma.stanford.edu/>

# Aplicaciones

Voz

Texto a voz  
Voz a texto  
VAD  
Identificación de Hablante.  
Reconocimiento de  
emociones

Música

Reconocimiento de acordes, notas, instrumentos, canciones, género.  
Clasificación de sonidos urbanos  
Clasificación de sonidos naturales  
Generación de música  
Detección y localización

Sonidos Ambientales



# Retos..



Fecha limite: Julio-Agosto

- Clásificación de genero
- Estimación de frecuencia fundamental
- Extracción de melodía
- Extracción de acordes
- Detección de Covers
- Transcripción de baterías



Fecha limite: 10 de junio

- Clasificación de escenas acústicas
- Localización de eventos y detección
- Detección de eventos sonoros en entornos domésticos
- Clasificación de sonidos urbanos
- Clasificación de sonidos con etiquetas(freesound y kaggle)
-

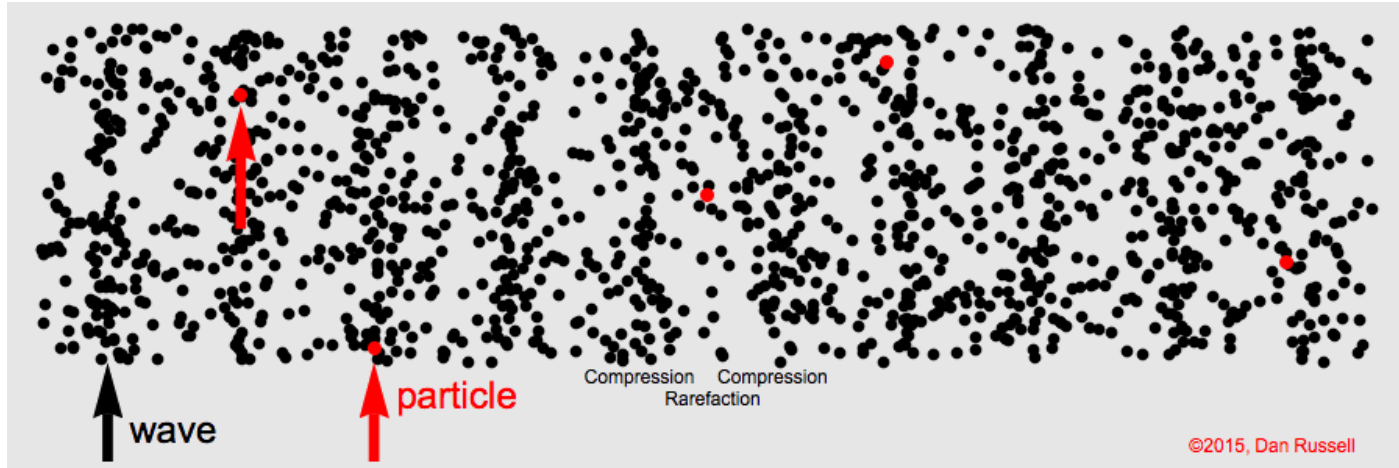
# Indice

- ¿Que es el sonido?
- Representaciones del sonido
- Modelos con Espectrograma de mel
- Modelos con forma de onda



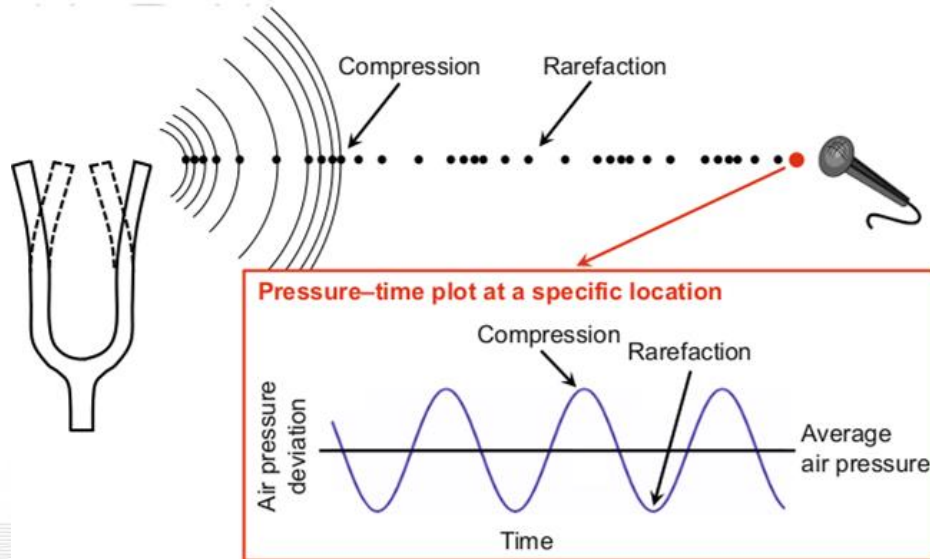
# ¿Que es el sonido?

•Pequeñas variaciones de la presión de un fluido a partir de su valor de equilibrio.



# ¿Como se representa el sonido en el computador?

• Las variaciones de presión capturadas por el Micrófono, se convierten en variaciones de voltaje, que posteriormente son digitalizadas para generar un archivo de audio.





# Representación tiempo-presión(raw audio)

•Parametros:

•Fs: Frecuencia de muestreo, máxima frecuencia que se puede capturar. 44.100 Hz calidad cd, típicamente 16.000Hz

•Bit depth: Relacionado con rango dinamico

$$20 \log(2^{16}) = 96dB$$

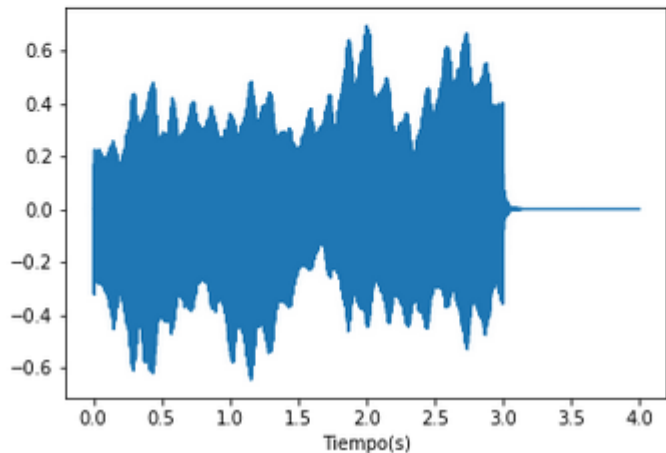


1 Second

# Lectura de archivos de audio en Python

```
In [4]: import matplotlib.pyplot as plt
        from scipy.io import wavfile

        fs, x = wavfile.read('organ_electronic_057-059-075.wav')
        x = x / (2**15 - 1)
        t = np.arange(0, x.size) / fs
        plt.xlabel('Tiempo(s)')
        plt.plot(t, x)
        plt.show()
```



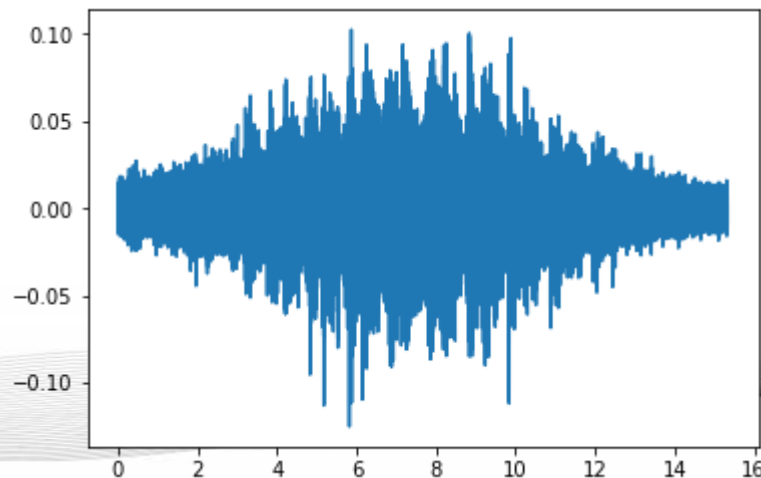
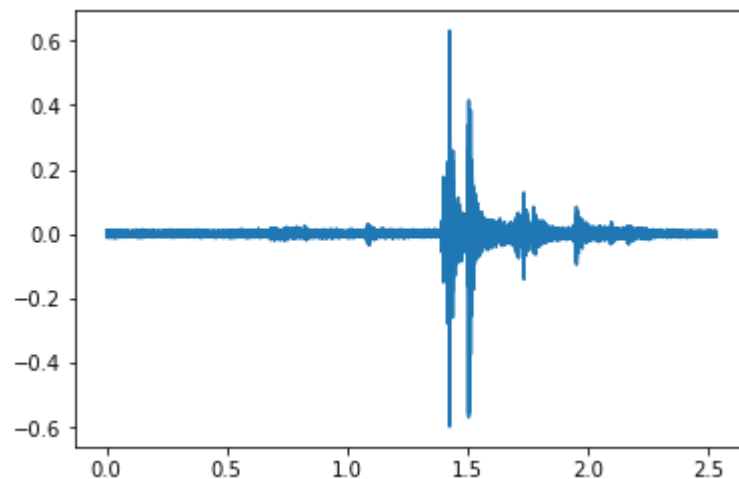
•Wavfile no lee archivos de 24 bits.

•Otras Librerías: Soundfile(libsoundfile), I

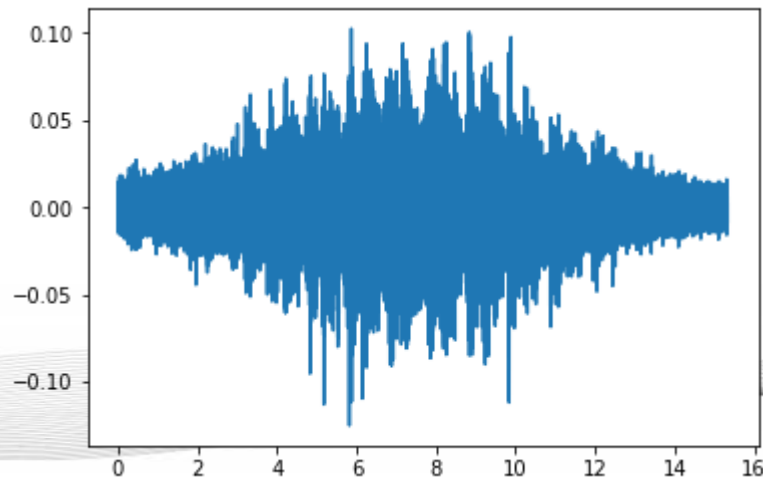
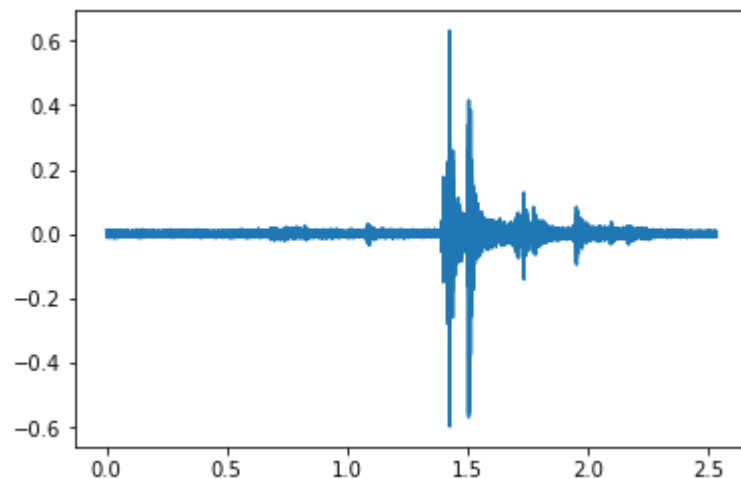
•Tensorflow 2.0

•tf.audio.decode\_wav

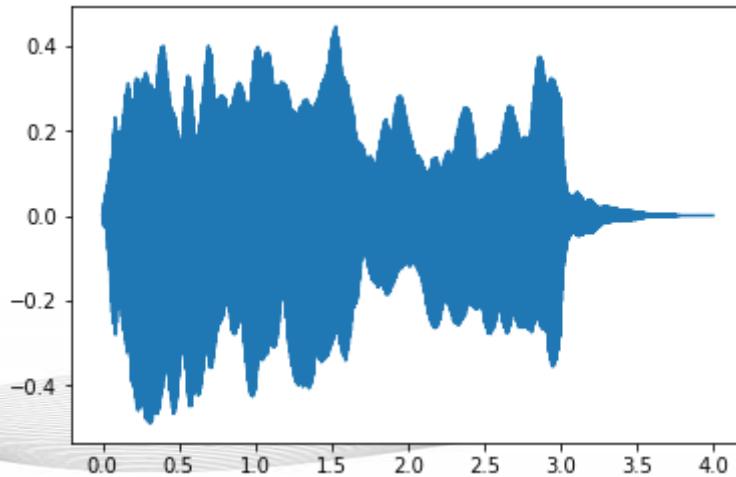
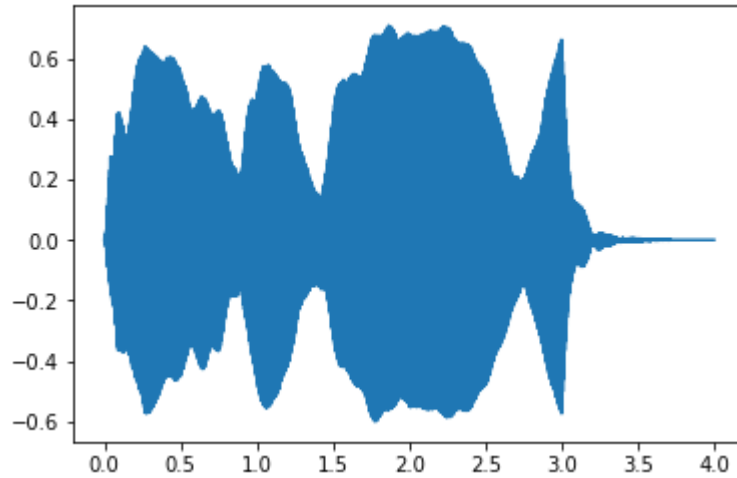
# A cual sonido corresponde?



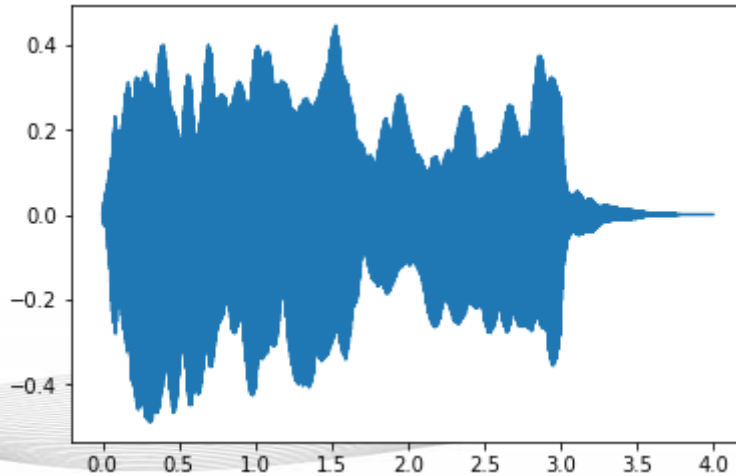
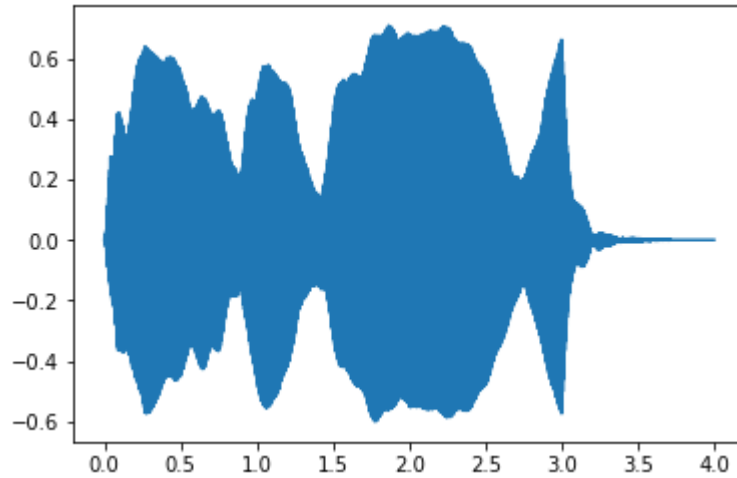
# A cual sonido corresponde?



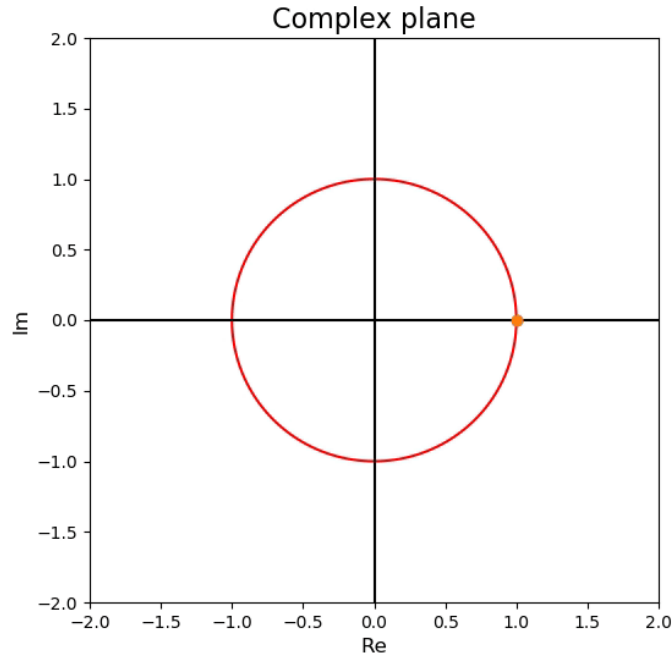
# A cual sonido corresponde?



# A cual sonido corresponde?



# Representación frecuencia(DFT)



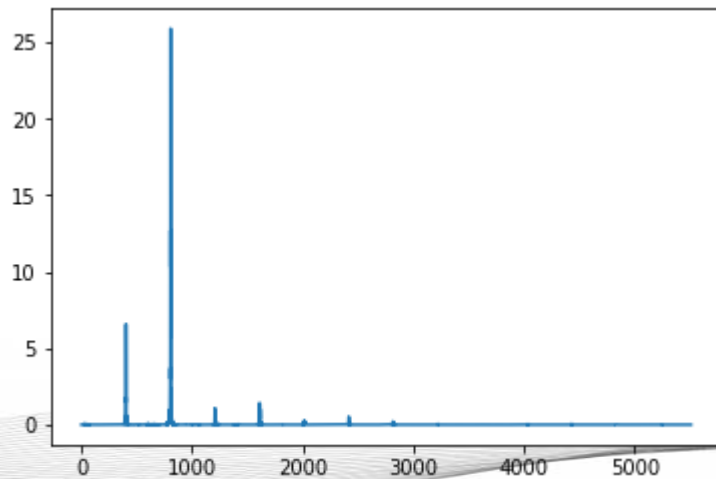
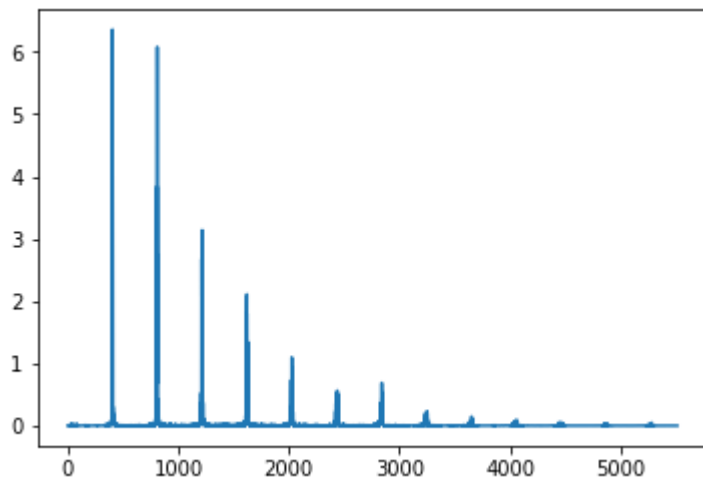
$$X(\omega_k) \triangleq \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi kn}{N}}, \quad k = 0, 1, 2, \dots, N-1.$$

• Parametros:

•  $N_{\text{fft}}$ : Tamaño de la DFT, Resolución en frecuencia  $\Delta f = F_s/N_{\text{FFT}}$

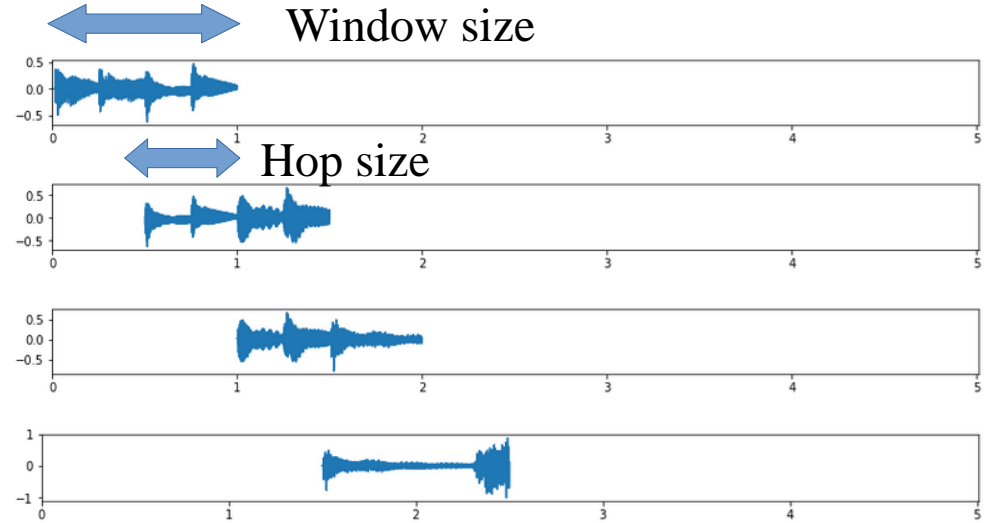
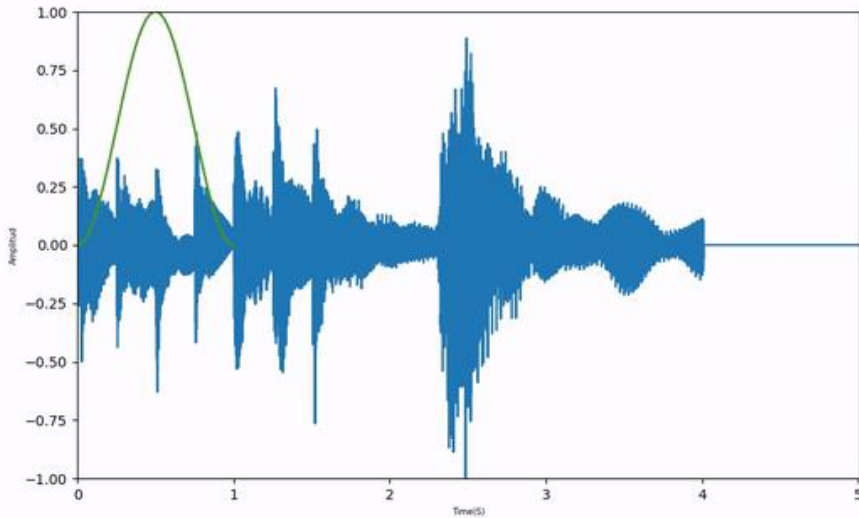
•

# Representación frecuencia(DFT)





# Los sonidos no son estáticos



El tamaño de la ventana debe ser lo suficientemente largo para representar el contenido en frecuencia.

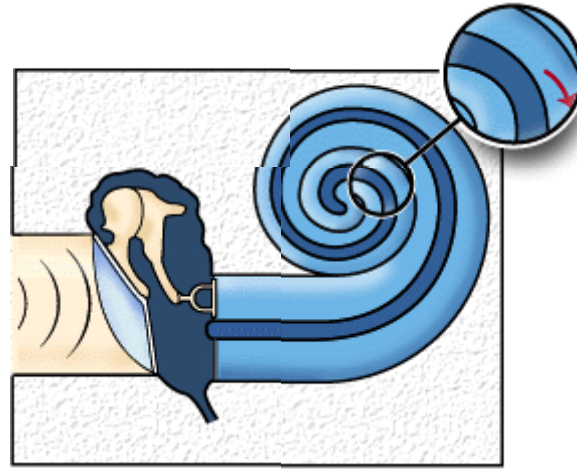
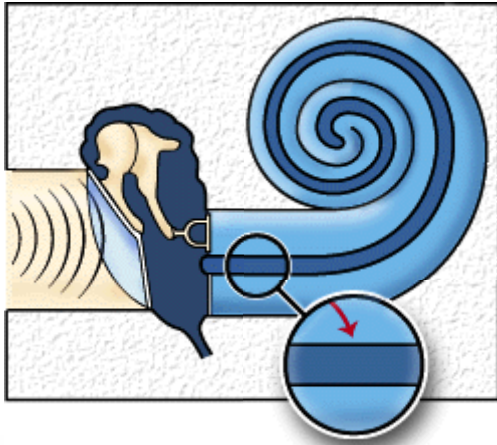
Sin embargo, no debe sobrepasar un tamaño que no permita ver las variaciones en el tiempo  
Del tono.

Tipicamente se usan ventanas de entre 20ms a 40ms dependiendo de la aplicación

# Audición

• Sentido que evolucionó para reconocer las amenazas en el entorno y tener una comunicación efectiva.

•



Fuente: <http://www.neuroreille.com/promenade/english/ear/fear.htm>

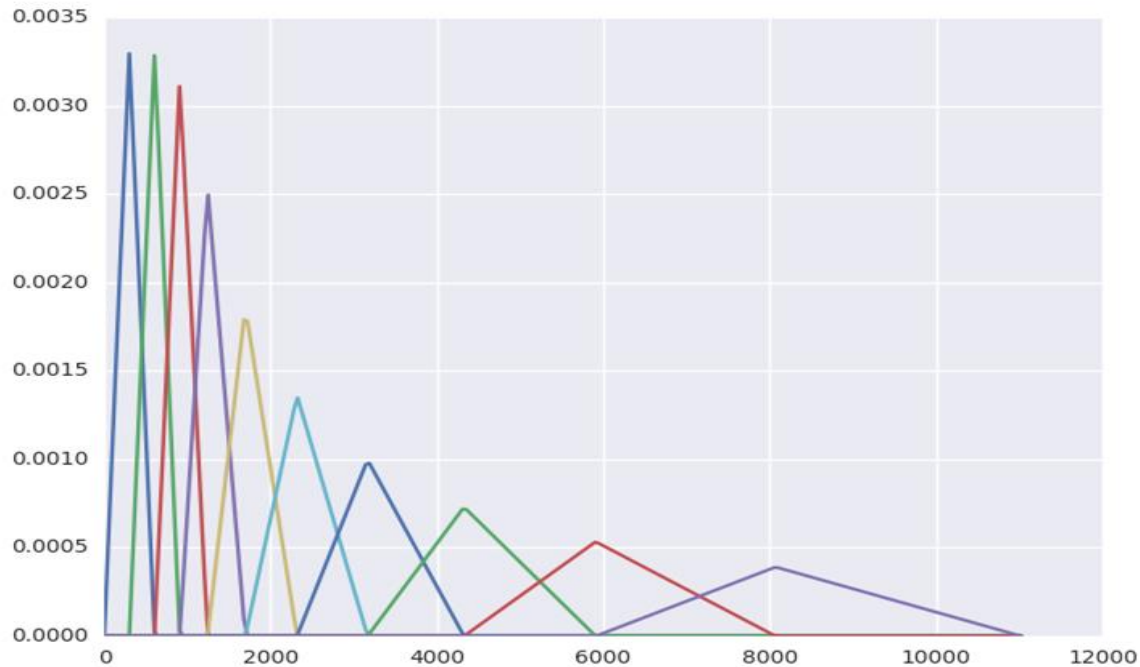
# Percepción de la frecuencia

80 hz

100Hz

3000Hz

3020Hz



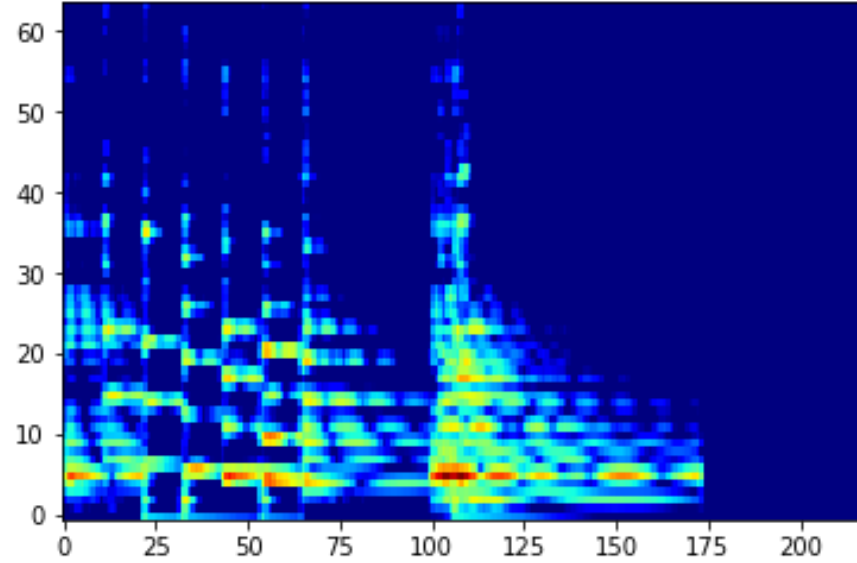
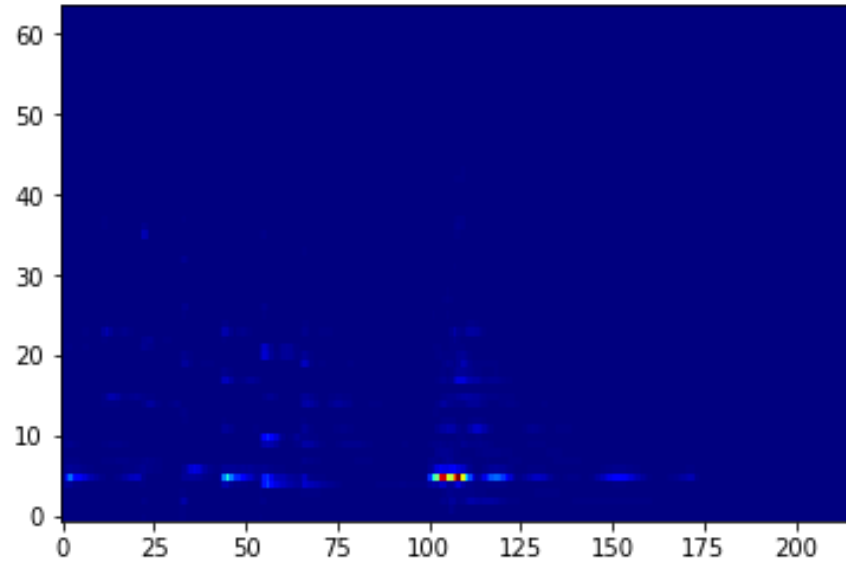
Mel filter bank

Librosa, Essentia(Python)

Auditory toolbox Matlab(Slaney)

HTK Hidden Markov Model Toolkit



# Percepción Amplitud




# Sistemas ganadores en el DCASE

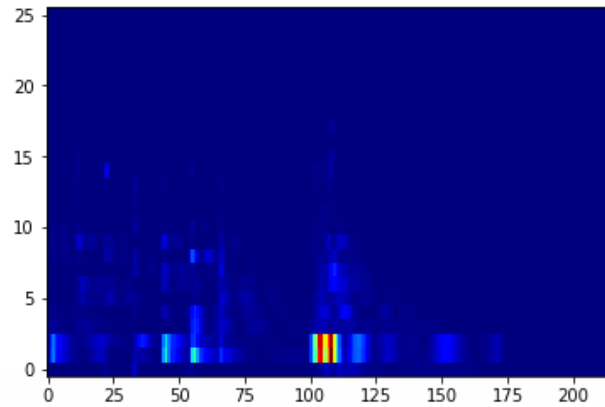
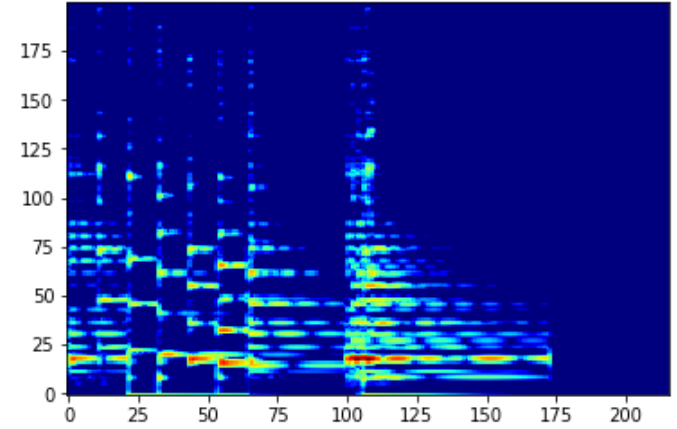
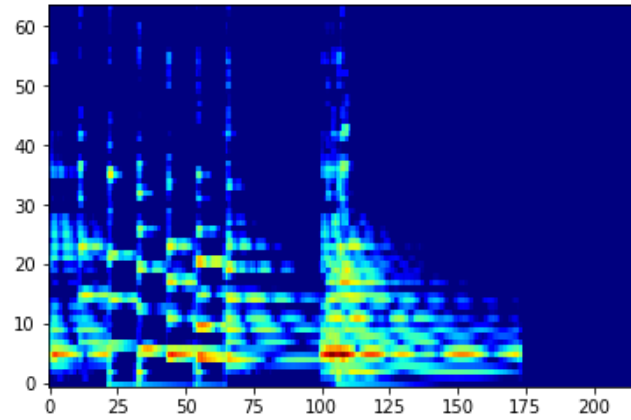
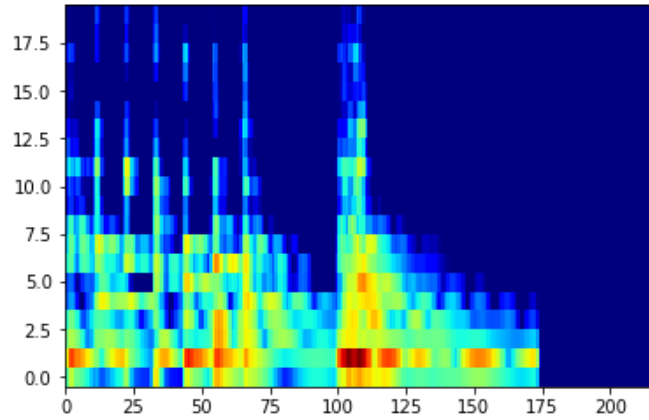
**General-purpose audio tagging of Freesound content with AudioSet labels**

**scene classification**

mAP@3  ▼	Acoustic features 
0.9538	<div><div>log-mel energies</div><div>waveform</div></div>
0.9518	<div><div>Perceptual weighted power spectrogram</div><div>Logarithmic-filtered log-spectrogram</div></div>
0.9512	<div><div>log-mel energies</div></div>
0.9506	<div><div>log-mel energies</div><div>waveform</div></div>
0.9498	<div><div>log-mel energies</div></div>
0.9496	<div><div>log-mel energies</div></div>

Features 
<div><div>log-mel energies</div></div>
<div><div>log-mel energies</div></div>
<div><div>log-mel energies</div></div>
<div><div>perceptual weighted power spectrogram</div><div>MFCC</div></div>
<div><div>perceptual weighted power spectrogram</div></div>

# ¿Cuantos Filtros?

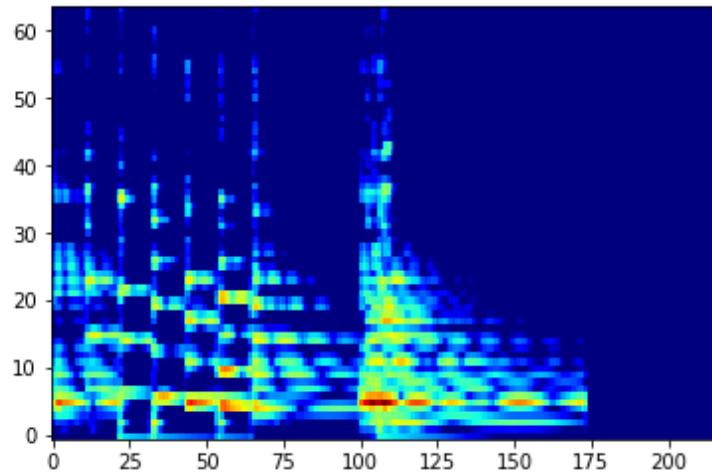


**THE DETAILS THAT MATTER: FREQUENCY RESOLUTION OF SPECTROGRAMS IN ACOUSTIC SCENE CLASSIFICATION** Detection and C

# Espectrograma de mel en Librosa

```
In [34]: mel = librosa.feature.melspectrogram(guitar_long,fs,n_fft=2048,hop_length=512,n_mels=64,fmax=4000)
log_mel = librosa.core.amplitude_to_db(mel)
plt.imshow(log_mel,origin='lower',aspect='auto',cmap='jet')
mel.shape
```

```
Out[34]: (64, 216)
```



# Base de Datos Audioset y Modelo VGGish

- Audioset es una base de datos de sonidos ambientales lanzada por Google en 2017.
- Contiene 2,084,320 ejemplos de clips de 10 segundos etiquetados en 527 categorías de eventos sonoros.
- Gemmeke, J. et. al., **AudioSet: An ontology and human-labelled dataset for audio events**, ICASSP 2017
- Hershey, S. et. al., **CNN Architectures for Large-Scale Audio Classification**, ICASSP 2017

<https://research.google.com/audioset/ontology/index.html>





# Formatos base de datos

- Archivo CSV que contiene los metadatos para descargar los sonidos desde youtube.

# Segments csv created Sun Mar 5 10:56:58 2017			
# num_ytids=2041789	num_segs=2041789	num_unique_labels=527	num_positive_labels=4020212
# YTID	tiempo_inicio	Tiempo_final	etiquetas
--1_cCGK4M	0	10	/m/01g50p,/m/0284vy3,/m/06d_3,/m/07jdr,/m/07rwm0c
--2_BBVHAA	30	40	/m/09x0r
--B_v8ZoBY	30	40	/m/04rlf
--EDNidJUA	30	40	/m/02qldy,/m/02zsn,/m/05zppz,/m/09x0r
--N4cFAE1A	21	31	/m/04rlf,/m/09x0r
--fcVQUf3E	30	40	/m/019jd,/m/07yv9
--g9OGAhwc	30	40	/m/04rlf,/m/0c1dj
--ITs1dxhU	30	40	/m/012f08,/m/07yv9,/m/0k4j,/t/dd00134
--mO--kRQk	30	40	/m/04rlf

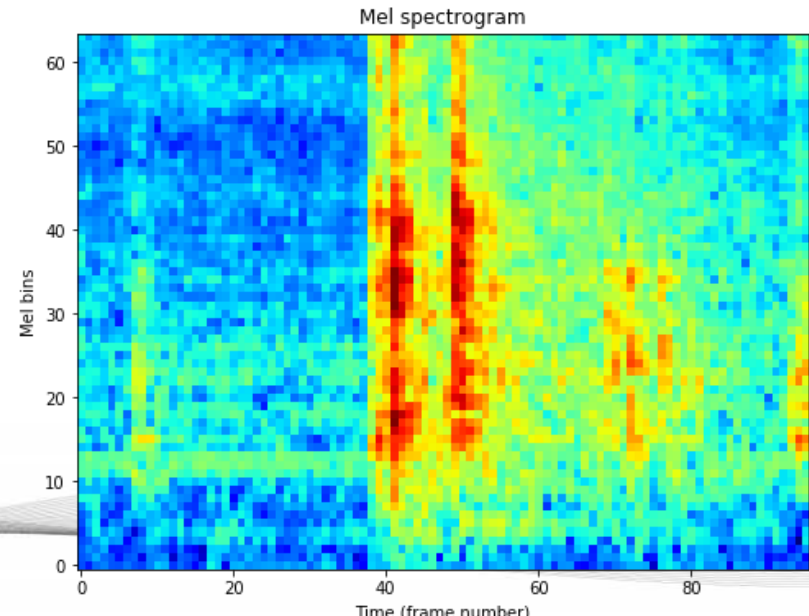
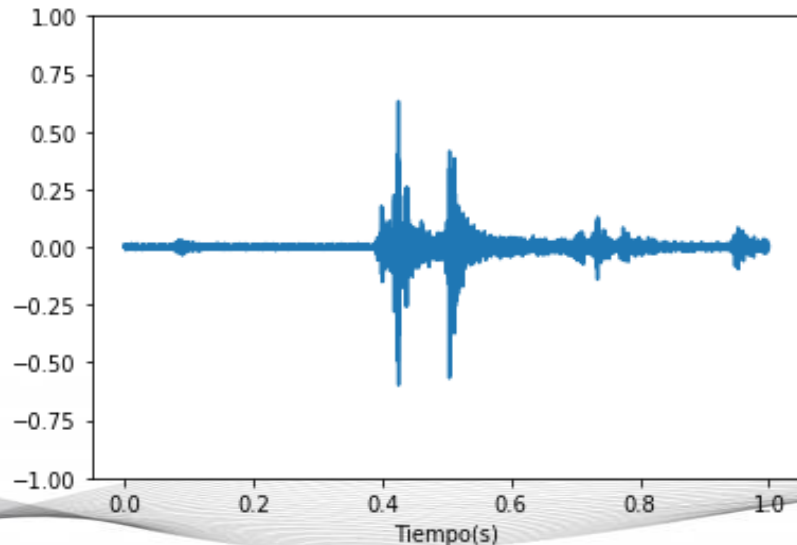
# Formato Embeddings

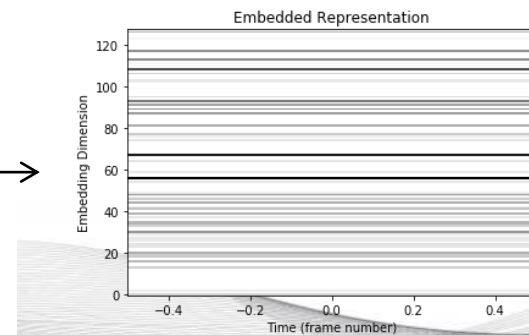
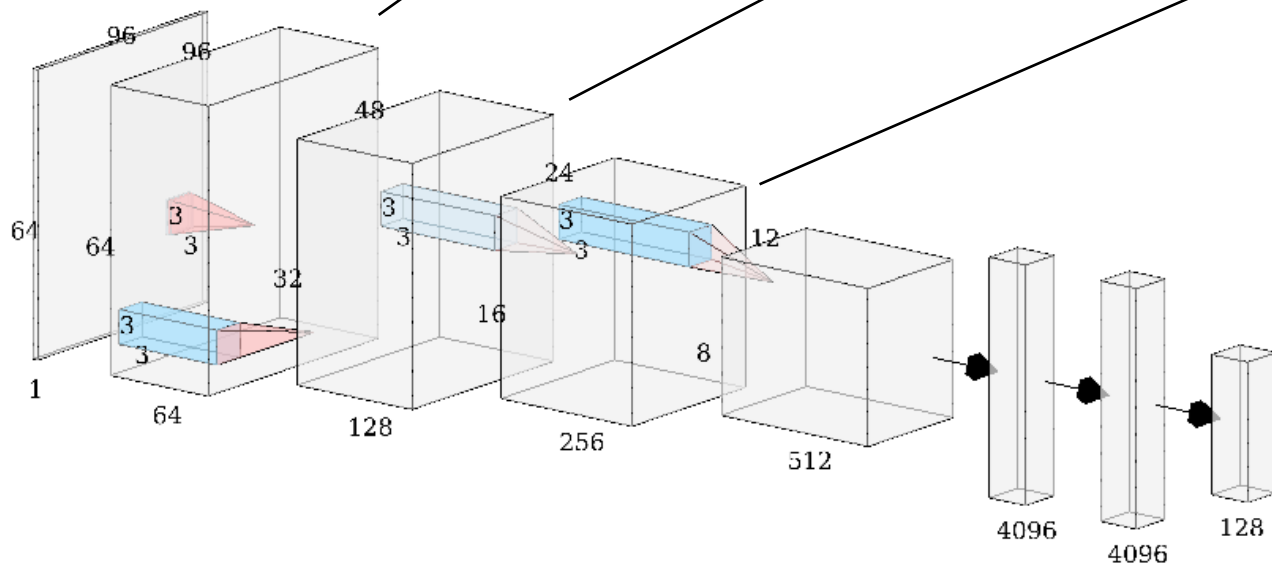
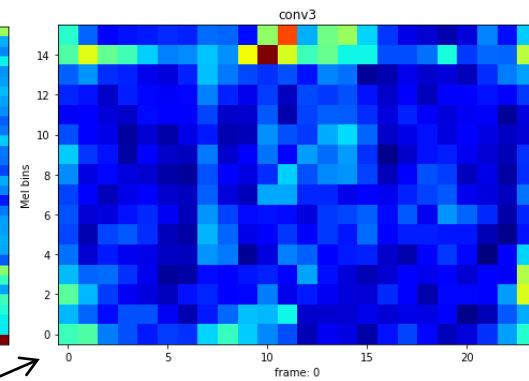
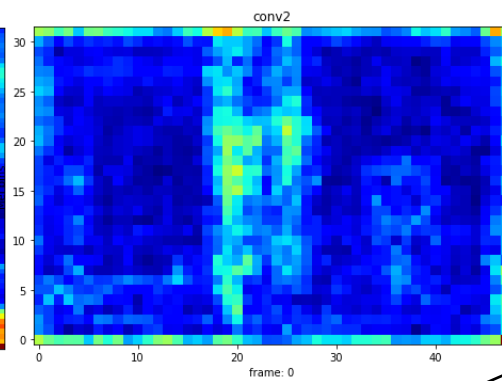
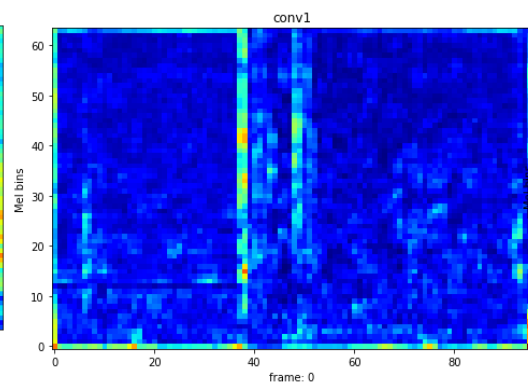
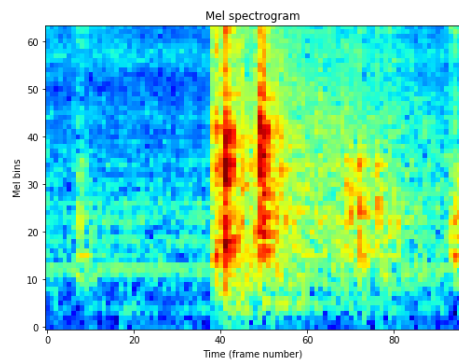
• Archivos en formato Tensorflow record con un vectores de características de 128D por cada segundo de audio. Estos vectores son obtenidos con el modelo VGGish que recibe como entrada espectrogramas de mel y genera esta compresión.

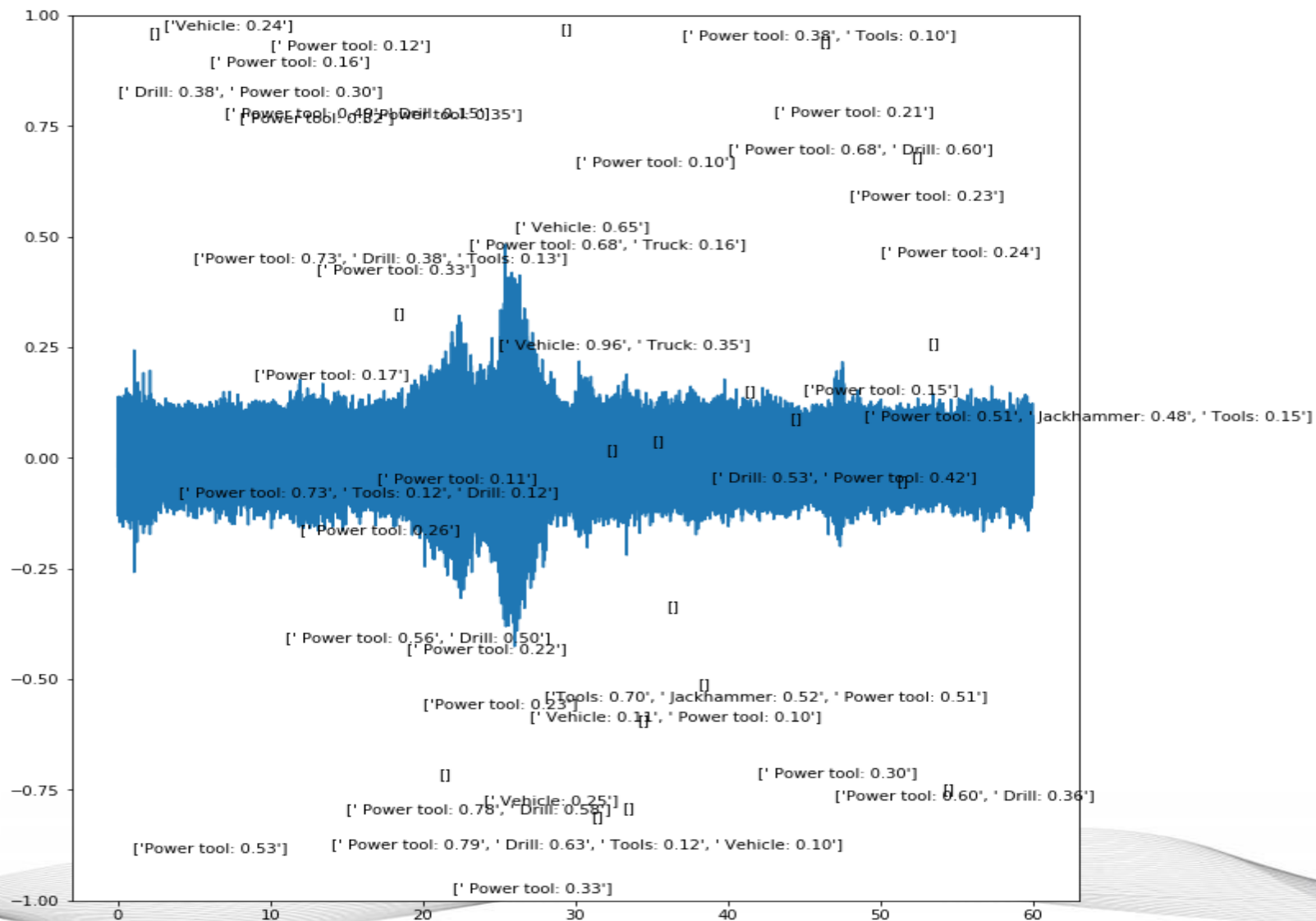
```
feature: {  
  key : "end_time_seconds"  
  value: {  
    float_list: {  
      value: 16.0  
    }  
  }  
}  
feature: {  
  key : "labels"  
  value: {  
    int64_list: {  
      value: [1, 522, 11, 172] # The meaning of the labels  
    }  
  }  
}  
feature_lists: {  
  feature_list: {  
    key : "audio_embedding"  
    value: {  
      feature: {  
        bytes_list: {  
          value: [128 8bit quantized features]  
        }  
      }  
      feature: {  
        bytes_list: {  
          value: [128 8bit quantized features]  
        }  
      }  
    }  
  }  
}
```

# Modelo VGGish

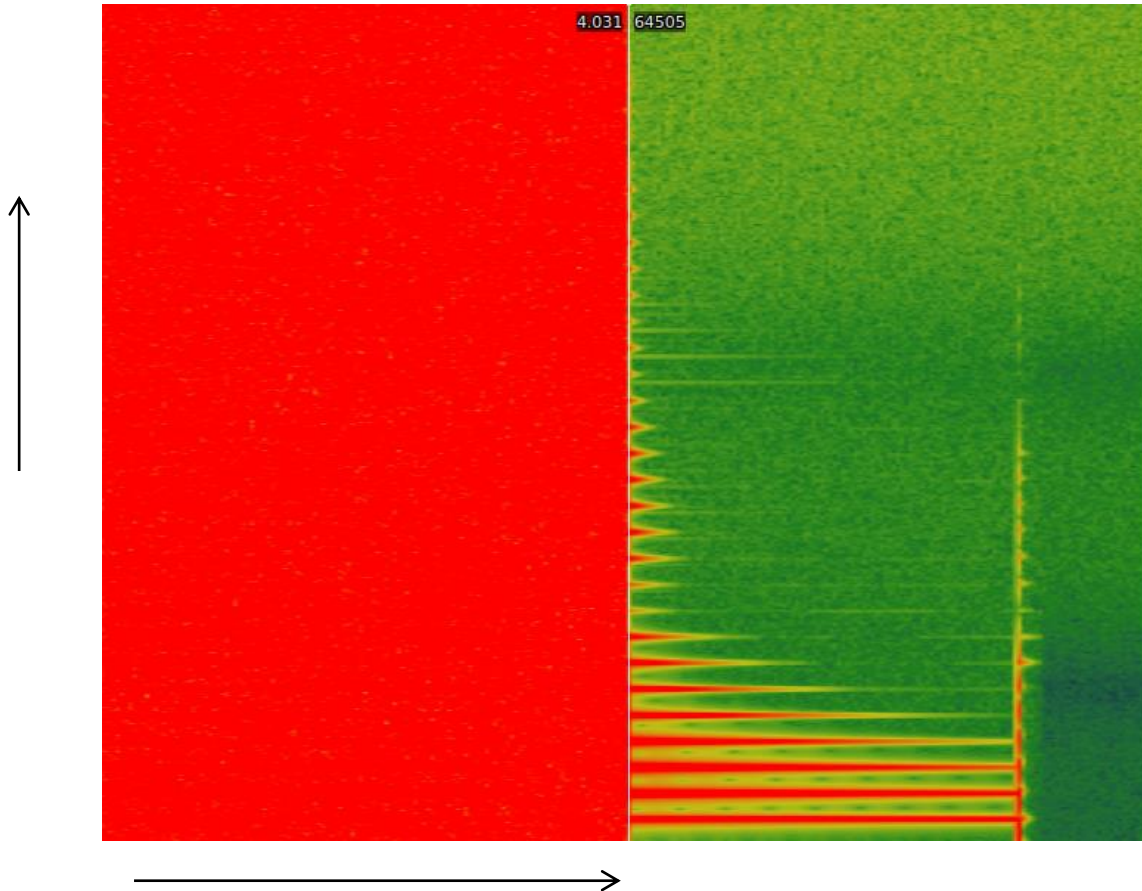
- La entrada al modelo VGGish son espectrogramas de mel de tamaño (96 frames x 64 mel bins).
- Los audios son resampleados primero a una frecuencia de 16Khz, luego se calcula una STFT con 25ms de tamaño de ventana y 10ms de hop size,
- frecuencia minima de 125hz y máxima de 7500hz







# Diferencia entre imagenes y sonido



Las dimensiones en un Espectrograma representan Conceptos diferentes.

Aunque puede existir invariancia En el eje temporal, esto no se Conserva en el eje de frecuencias. (filtros rectangulares)

# Técnicas para aumentar datos

- \* Transposición de tono
- \* Reverberación: Convolución con respuestas al impulso de diferentes ambientes
- \* Ruido de fondo: añadir ruido de ambientes reales, no solo blanco o rosa.
- \* Dispositivo de captura: Convolución con respuestas al impulso de grabadoras
- \* Posición del microfono

# Usando Directamente la forma de onda(raw-audio)

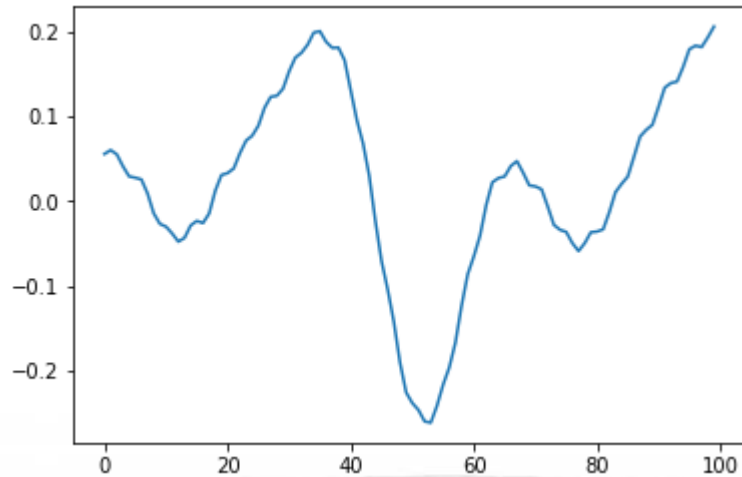
- ¿Porque no usar directamente una RNN?
- Un frame de 25ms de audio corresponde a 400 muestras con una frecuencia de muestreo de 16.000Hz
- Modelar la estructura temporal de un sonido que dura varios segundos requiere usar secuencias muy largas
- ¿Que usar entonces?
- Trabajos previos hacen uso de redes convolucionales en una dimensión con algunas modificaciones





# Usando Raw Audio

0.1	0.2	0.6	-0.3	0.4	0.3	-0.7	0.4	0.1	0.3
-----	-----	-----	------	-----	-----	------	-----	-----	-----



Pre-Procesamiento: Pasar a Mono,  
Resamplear

# Usando Raw Audio 1D

0.1	0.2	0.6	-0.3	0.4	0.3	-0.7	0.4	0.1	0.3
-----	-----	-----	------	-----	-----	------	-----	-----	-----

W1	W2	W3
----	----	----



Kernel Size 3



# Usando Raw Audio 1D

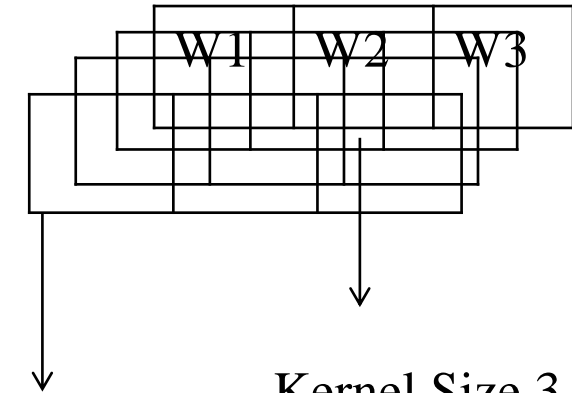
0.1	0.2	0.6	-0.3	0.4	0.3	-0.7	0.4	0.1	0.3
-----	-----	-----	------	-----	-----	------	-----	-----	-----

(batch, steps, channels)

Salida:  $\text{seq len} - \text{kernel size} + 1$

Kernel Size 3

Filters 4



# Usando Raw Audio 1D

0.1	0.2	0.6	-0.3	0.4	0.3	-0.7	0.4	0.1	0.3
-----	-----	-----	------	-----	-----	------	-----	-----	-----

W1	W2	W3
----	----	----

W1	W2	W3
----	----	----



Stride 1



# Usando Raw Audio 1D

0.1	0.2	0.6	-0.3	0.4	0.3	-0.7	0.4	0.1	0.3
-----	-----	-----	------	-----	-----	------	-----	-----	-----

W1	W2	W3
----	----	----

W1	W2	W3
----	----	----

W1	W2	W3
----	----	----



# Usando Raw Audio 1D

0.1	0.2	0.6	-0.3	0.4	0.3	-0.7	0.4	0.1	0.3
-----	-----	-----	------	-----	-----	------	-----	-----	-----

W1	W2	W3
----	----	----

W1	W2	W3
----	----	----

W1	W2	W3
----	----	----

W1	W2	W3
----	----	----



# Usando Raw Audio 1D

0.1	0.2	0.6	-0.3	0.4	0.3	-0.7	0.4	0.1	0.3
-----	-----	-----	------	-----	-----	------	-----	-----	-----

W1	W2	W3
----	----	----

W1	W2	W3
----	----	----

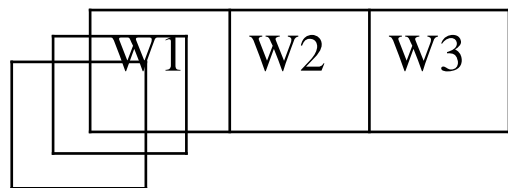
W1	W2	W3
----	----	----

W1	W2	W3
----	----	----

W1	W2	W3
----	----	----

# Using Raw Audio 1D

0	0.1	0.2	0.6	-0.3	0.4	0.3	-0.7	0.4	0.1	0.3	0
---	-----	-----	-----	------	-----	-----	------	-----	-----	-----	---



Padding

Output = seqLen



# CREPE(A Convolutional Representation for Pitch Estimation)

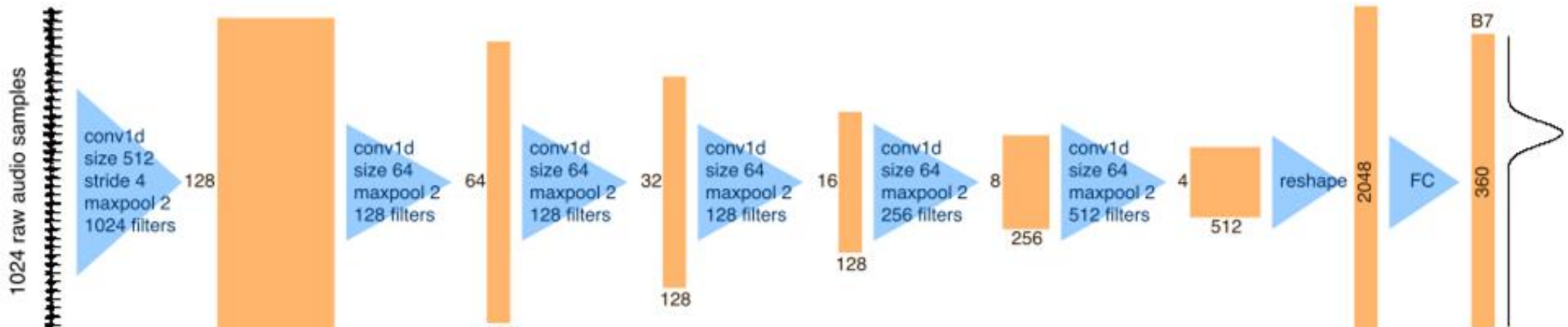
Regresión o clasificación?

<https://marl.github.io/crepe/>

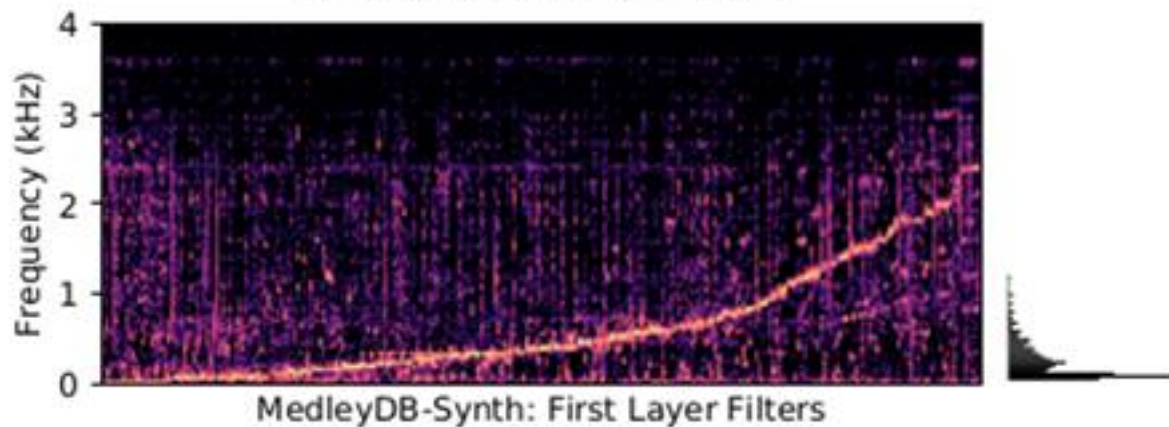
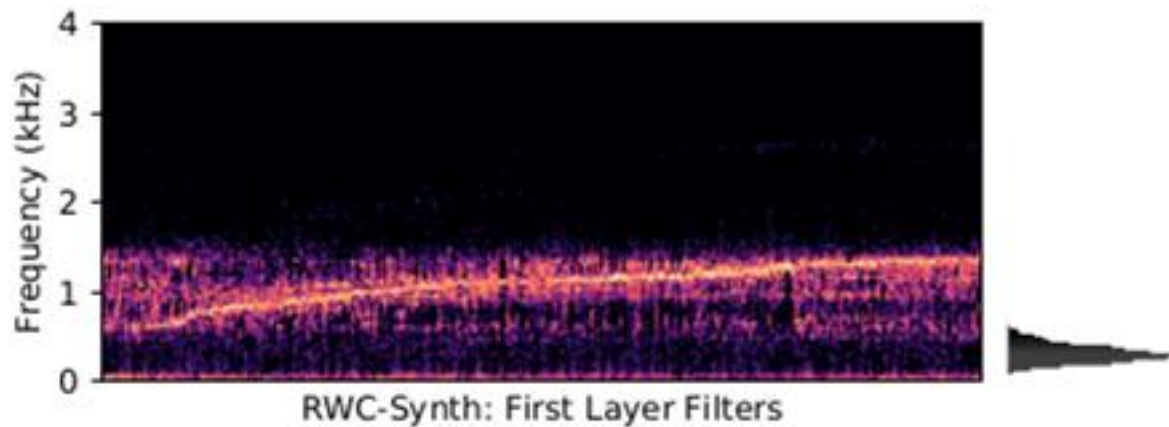
Entrada: 1024 Muestras - 60ms

Salida: 360 vector de cents

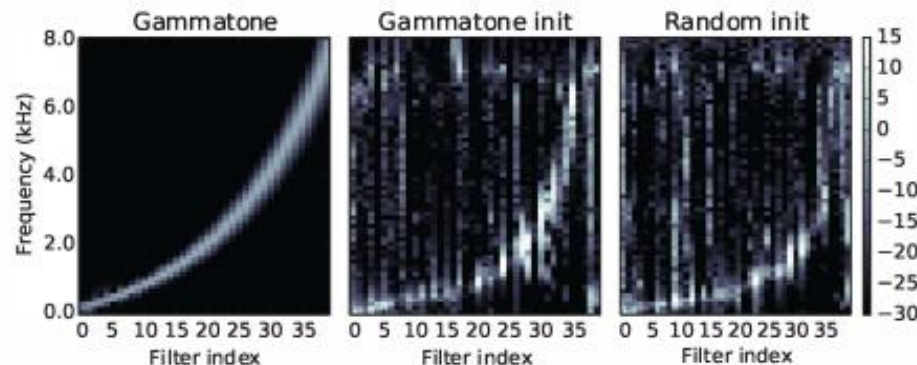
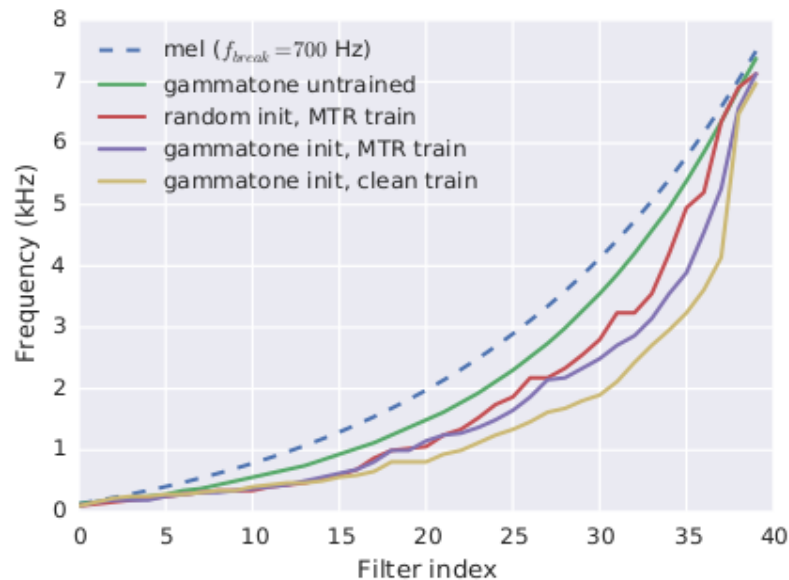
[Jong Wook Kim](#)



# Espectro de los filtros en la primera capa



# Espectro de los filtros en la primera capa



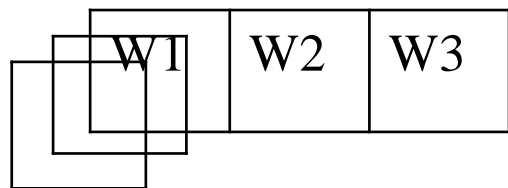
Hoshen, et al. “**Speech acoustic modeling from raw multichannel**

T. N. Sainath, et al. “**Learning the speech front-end with raw waveform cldnns,**” in Interspeech, 2015.

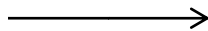
Zeghidour, Neil et al. “**End-to-End Speech Recognition from the Raw Waveform.**” Interspeech (2018).

# Usando Raw Audio 1D

	0.1	0.2	0.6	-0.3	0.4	0.3	-0.7	0.4	0.1	0.3	
--	-----	-----	-----	------	-----	-----	------	-----	-----	-----	--



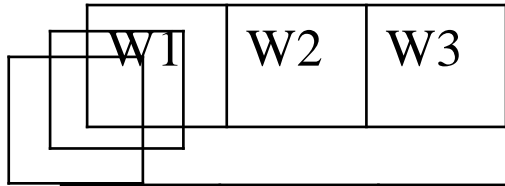
W1		W2		W3	
----	--	----	--	----	--



Dilatación

# Usando Raw Audio 1D

	0.1	0.2	0.6	-0.3	0.4	0.3	-0.7	0.4	0.1	0.3	
--	-----	-----	-----	------	-----	-----	------	-----	-----	-----	--



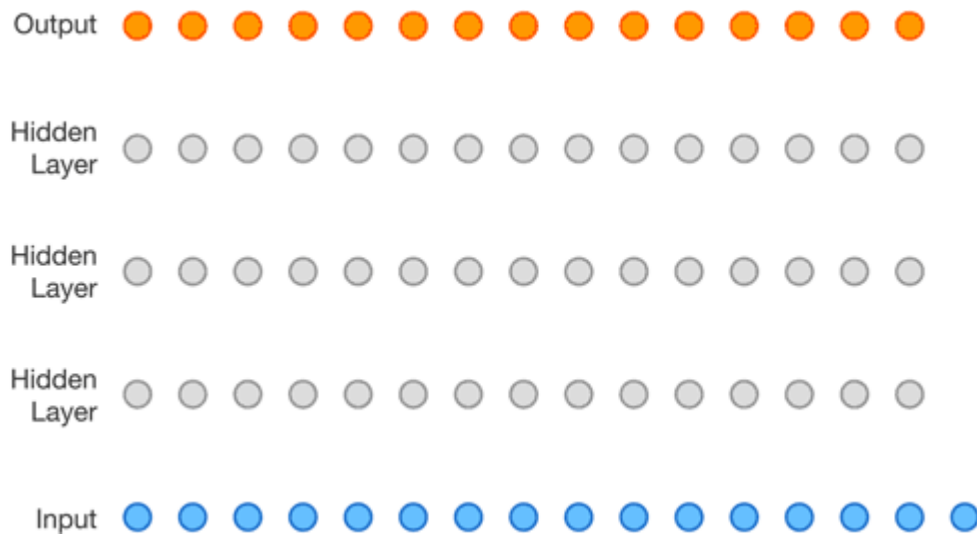
W1		W2		W3	
----	--	----	--	----	--

W1				W2				W3
----	--	--	--	----	--	--	--	----

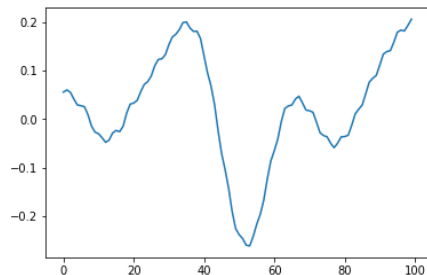
# Wavenet

Modelo generativo de audio, en donde cada muestra depende de las anteriores.

$$p(\mathbf{x}) = \prod_{t=1}^T p(x_t \mid x_1, \dots, x_{t-1})$$



# Wavenet



0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	1
0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	1	0	0	0

255

16 bit = 65,536

8 bit = 255

Compresión de amplitud  
U-law 8-bit

One hot encoded

samples

# Recursos

<https://musicinformationretrieval.com/index.html>

Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications

Computational Analysis of Sound Scenes and Events, Tuomas Virtanen



# Muchas Gracias por su atención

## Contacto:

[jose091@gmail.com](mailto:jose091@gmail.com)

[jose091@ccrma.stanford.edu](mailto:jose091@ccrma.stanford.edu)

•<https://www.linkedin.com/in/jose-o-giraldo/>

