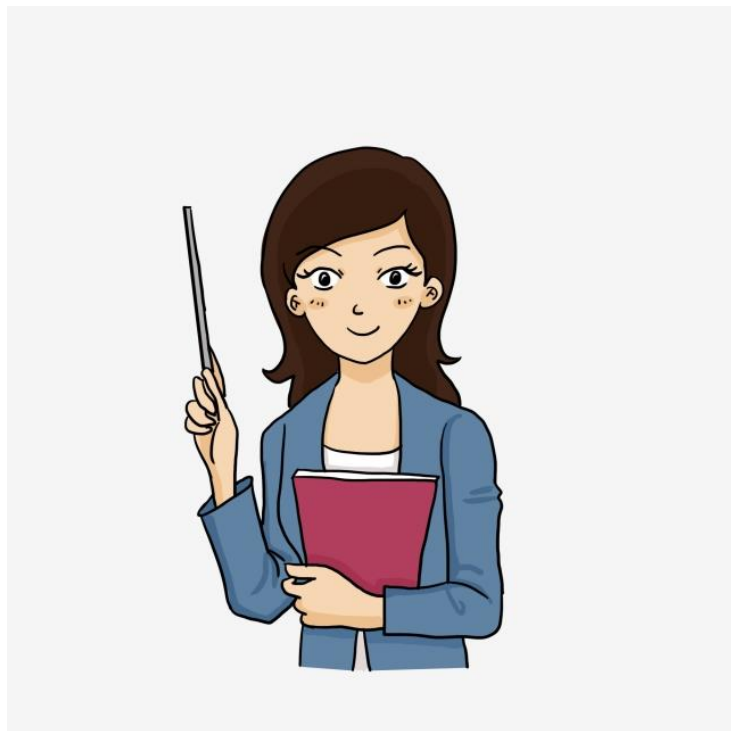


語音識別



老師的教導



媽媽的嘮叨



朋友的交流

語言是資訊的載體

智慧語音技術：使機器像人一樣“能聽會說”的技術，包括語音合成、語音辨識和自然語言理解等。



A、聽歌識曲、哼唱識別







B、同步速記



D、地圖導航明星播報



E、翻譯機

						
年份	1995年	1998年	1999年	2001年	2013年	
自然度	<3.0	3.0	3.5	3.8	4.5	

C、語音合成

STOP

原文：9，這是喬丹參加
1984年奧運會和1992年
奧運會時的球衣號碼.....

- 回顧
 - 聲波的產生和傳播原理

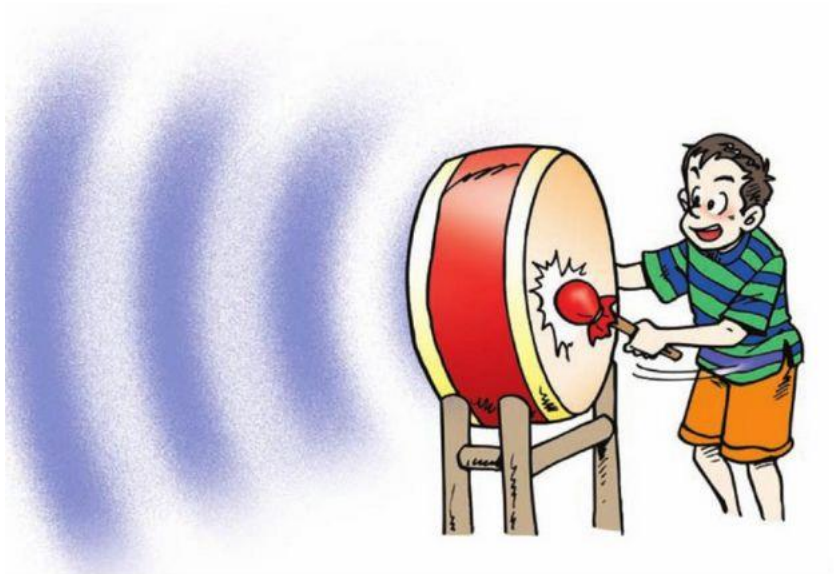


💡 思考與討論

- 怎樣讓這把鋼尺發聲？



- 回顧
 - 聲波的產生和傳播原理



振動

+

介質

+

人耳

=

感知

- 聲音的三個特性

- 音調

- 代表聲音的高低
 - 決定于聲源的振動頻率
 - 頻率：物體在1秒鐘內震動的次數，單位是赫茲（Hz）
 - 頻率越高，音調就越高，聽起來越尖銳

- 聲音的三個特性

- 響度

- 代表聲音的強弱
 - 決定于聲源的振動幅度，即振幅
 - 振幅越大，響度就越大，聲音聽起來越大

- 聲音的三個特性
 - 音色
 - 代表聲音的特色
 - 決定於聲源本身材料、結構等特性



尼龙弦

古典吉他



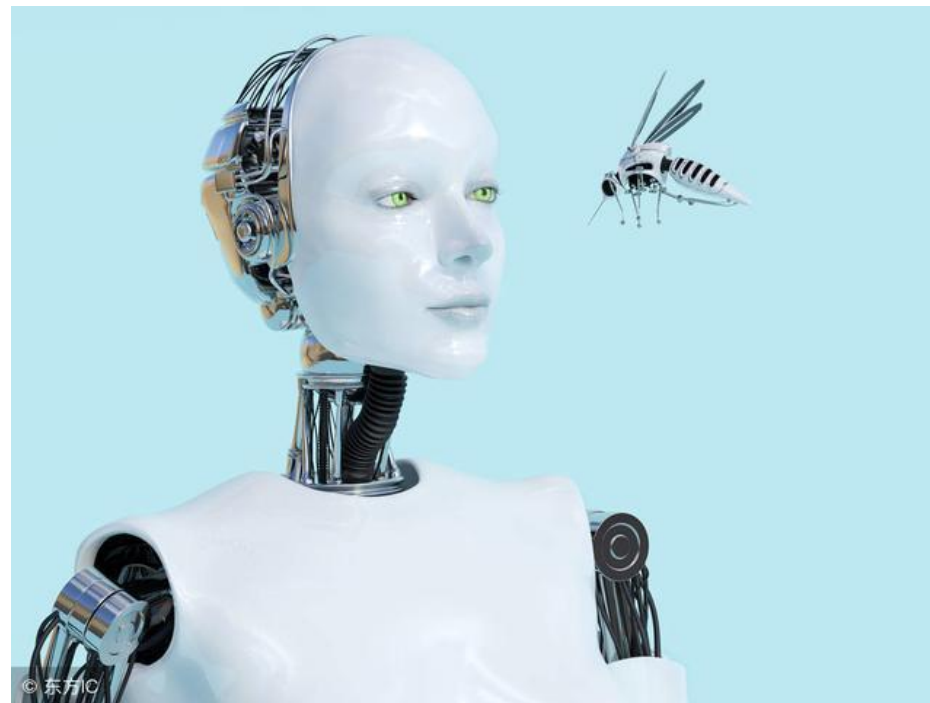
金属弦

民谣吉他

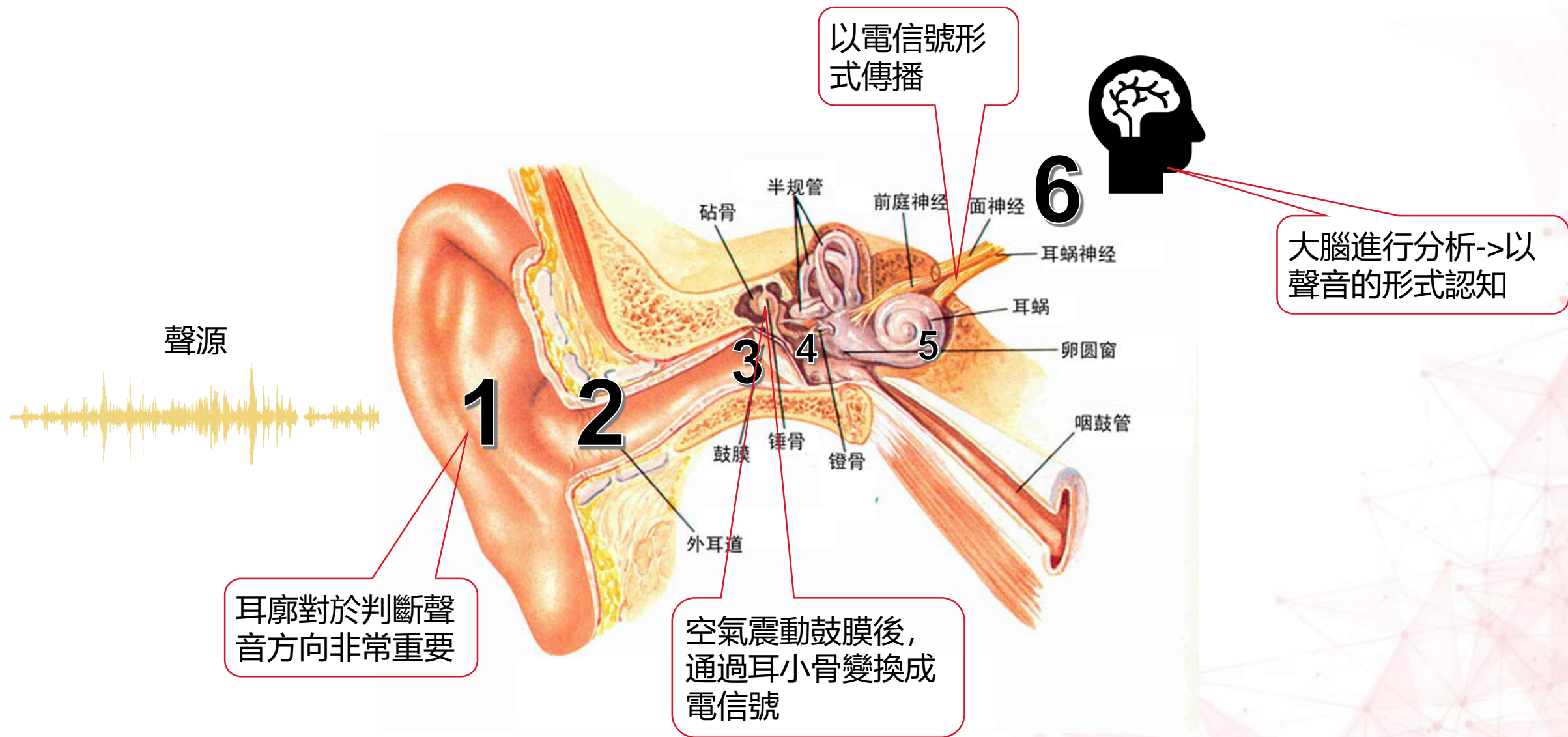
不同樂器的音色不同

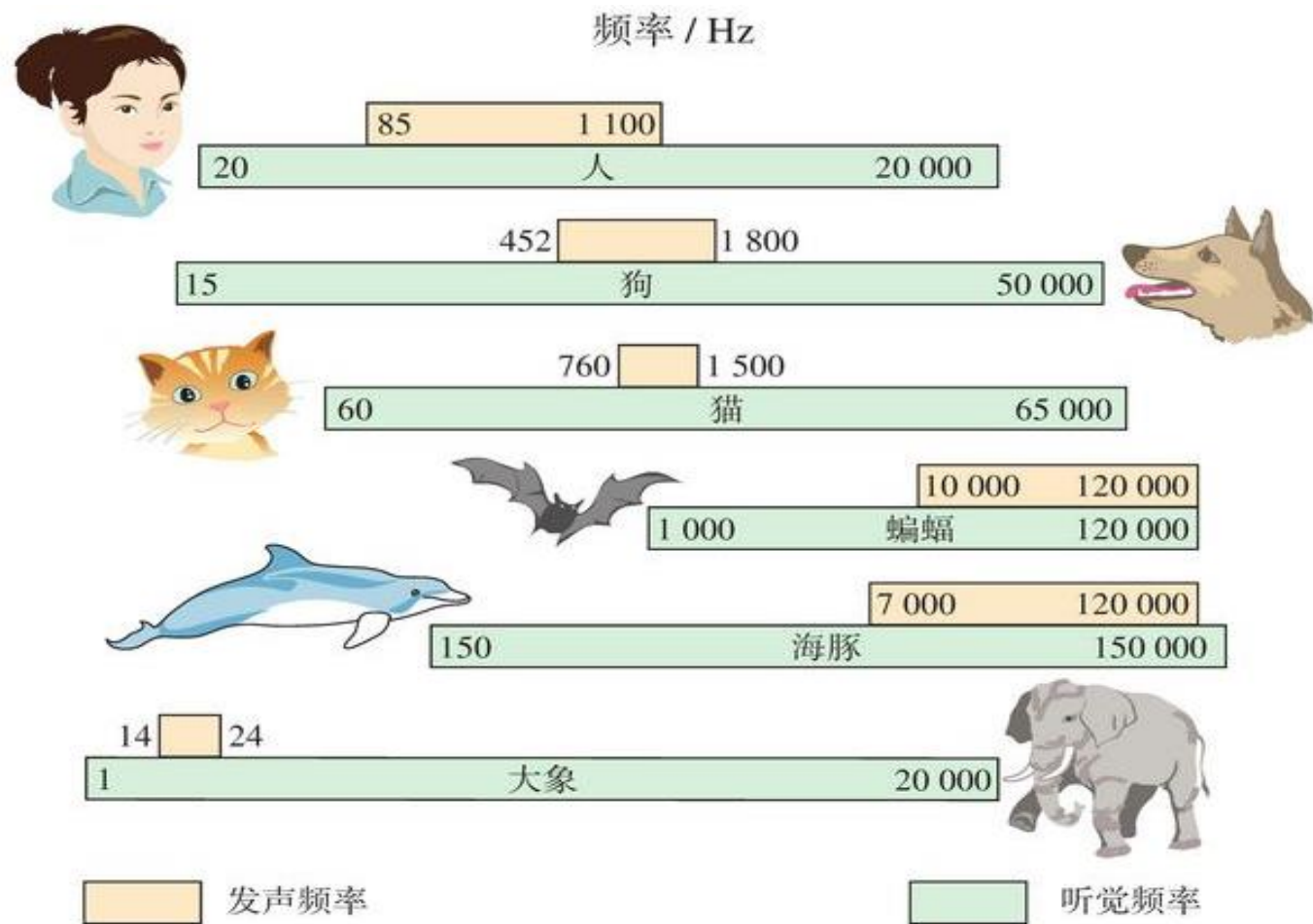
Q: 如何將聲波轉換為電腦可以存儲和處理的音訊檔
(MP3) ?

提示: 連續 -> 離散



聽覺的原理





人和动物的发声、听觉频率范围

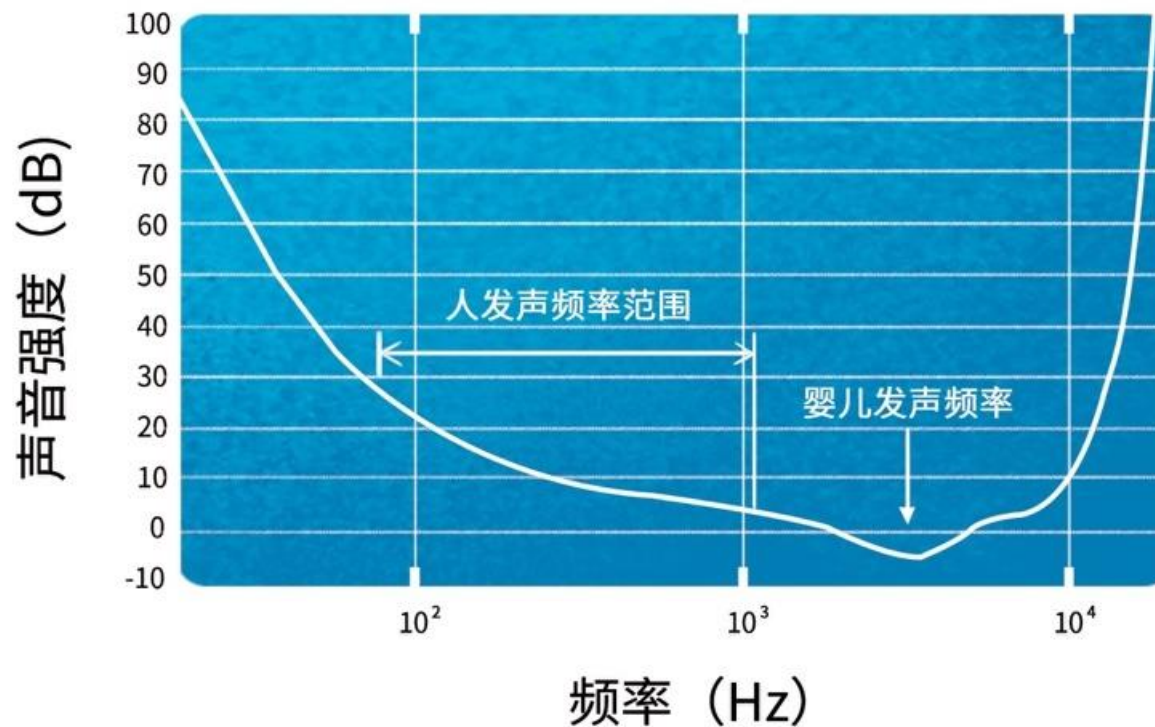
- 頻率

- 定義：發聲物體在一秒內振動的次數
- 人耳對不同頻率聲音有不同的敏感程度

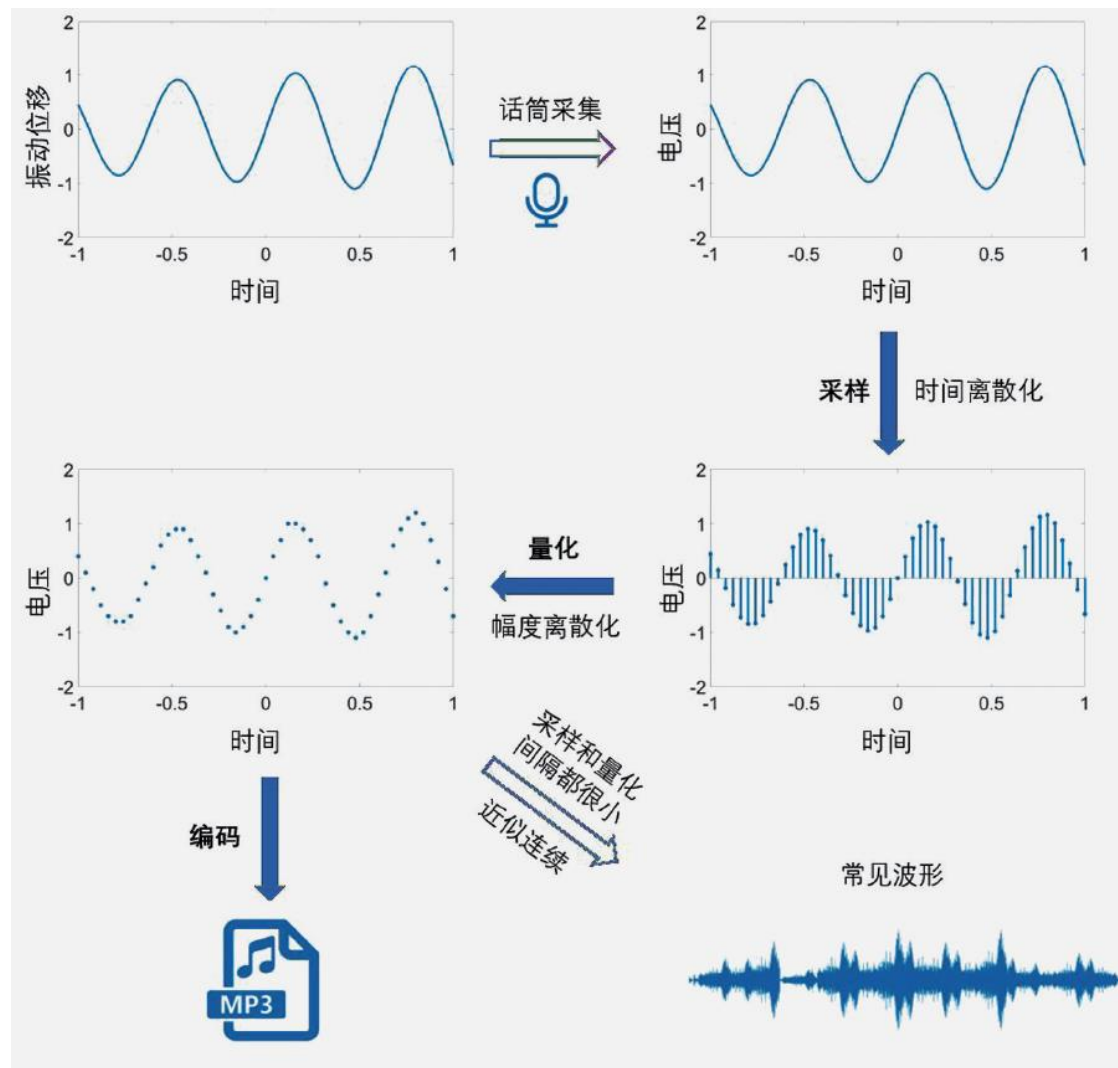
- 一些奇妙的事實

- 對低頻和高頻都不敏感
- 對嬰兒聲頻最為敏感
- 對人聲頻率相對敏感

人耳聽覺對不同頻率聲音的敏感程度



如何數位化表達聲音



聲音數位化過程

採樣：時間離散化，採樣頻率

採樣頻率：44.1kHz (MP3音質)，
22.05kHz (FM音質)

量化：幅度離散化，量化位元數

一般量化位數：8bit, 16bit

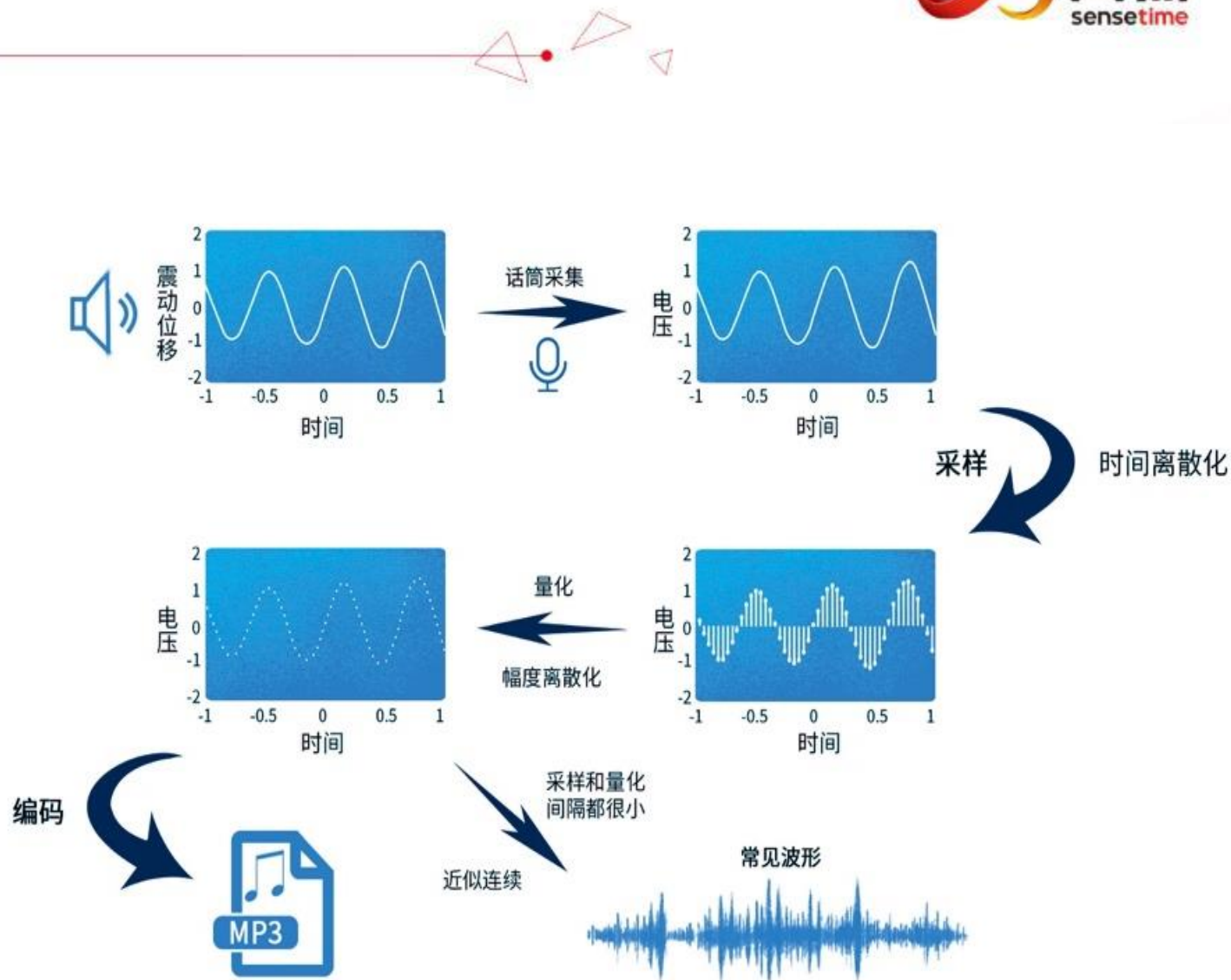
編碼：便於存儲和傳輸，編碼演算法

聲道數：2bps (雙聲道)

碼率 = 採樣頻率 * 量化位數 * 聲道

如何數位化表達聲音

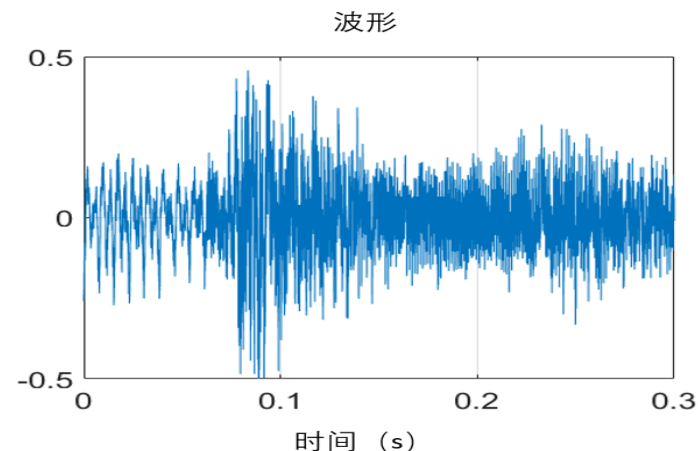
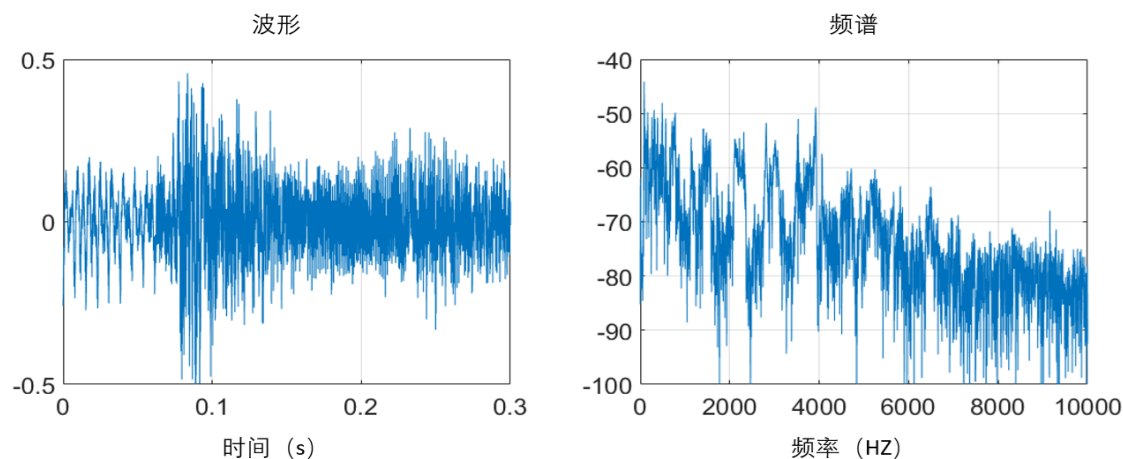
- 採樣 **橫向離散**
 - 將時間上的連續信號離散化
 - 取樣速率：每秒採集聲音樣本數
- 量化 **縱向離散**
 - 將振幅離散化成一個個資料點
- 編碼
 - 進行格式轉換，去除冗餘信息



如何數位化表達聲音

波形

電腦裡面的音訊檔描述的實際上是一系列按時間先後順序排列的資料點, 所以也被稱為時間序列(time series), 把它視覺化出來就是我們常見的波形(waveform)。



頻譜(frequency spectrum)

頻譜的橫坐標代表頻率, 縱坐標表示頻譜幅度, 其含義是相應頻率的聲音所對應的振幅, 單位與波形的縱坐標相同。

因為一段音訊中不同頻率的聲音強度相差很大, 所以頻譜幅度通常使用對數座標, 即振幅每相差10倍, 頻譜幅度相差20。

頻譜圖反映了不同頻率的聲音所占能量的多少, 我們通常只關注頻譜幅度的相對大小。

SenseStudy課程平臺 “人工智能入門（下）”：實驗8 – 1 声音的特性与存储

```
url = "http://sensestudy-server/api/resource/public/accountstorage-objs/6ada5e27-552a-4437-a5cd-b0aed638e0bb/speech.wav"
```

```
my_audio = load_audio(url)
```

```
print(my_audio.sample_width)  
print(my_audio.channels)  
print(my_audio.frame_rate)
```

```
fig() + audio(url, "wave")
```



SenseStudy課程平臺 “人工智能基礎 ”： 實驗4 – 1 观察声音的波形，理解声音的数字化

```
gtzan = data.get('gtzan')  
  
music, label = gtzan[10]  
  
fig() + plot(music, type='audio')  
print(label)  
  
segment = music.cut(0.76, 0.78)  
  
fig() + plot(segment, type='waveform')  
  
segment_small = music.cut(0.76, 0.761)  
fig() + plot(segment_small, type='waveform')  
  
print(music.sample_rate)
```



如何從數位化表達中提取聲音的重要特徵

樂音三要素

響度：最直觀的樂音要素, 代表聲音的強弱。

音調：表示聲音調子的高低, 對應基音頻率的高低。

音色：是一種更複雜的特徵, 即便是相同的音調和響度, 用不同的樂器演奏或者不同的人來演唱都有不同的聽覺效果。泛音中各個頻率成分對應的幅度各有不同, 造就了獨特的聽覺感受

波形的幅度反映了響度

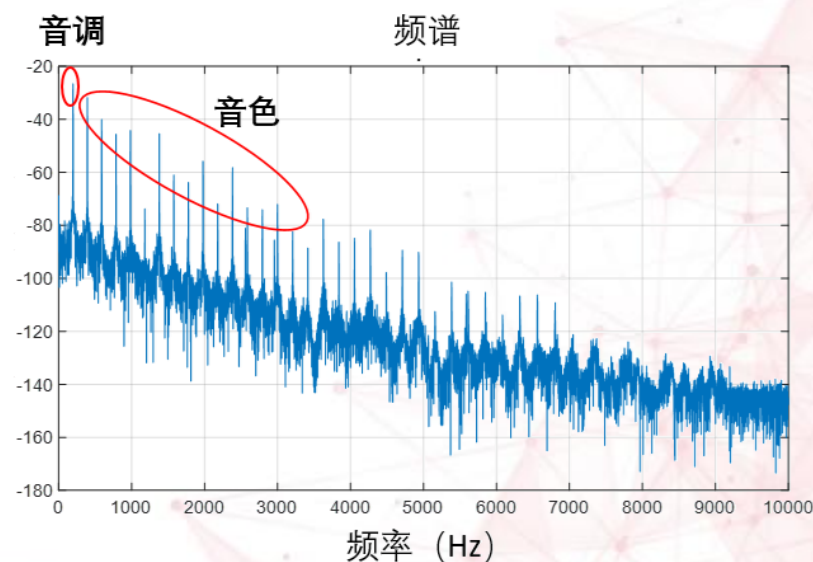
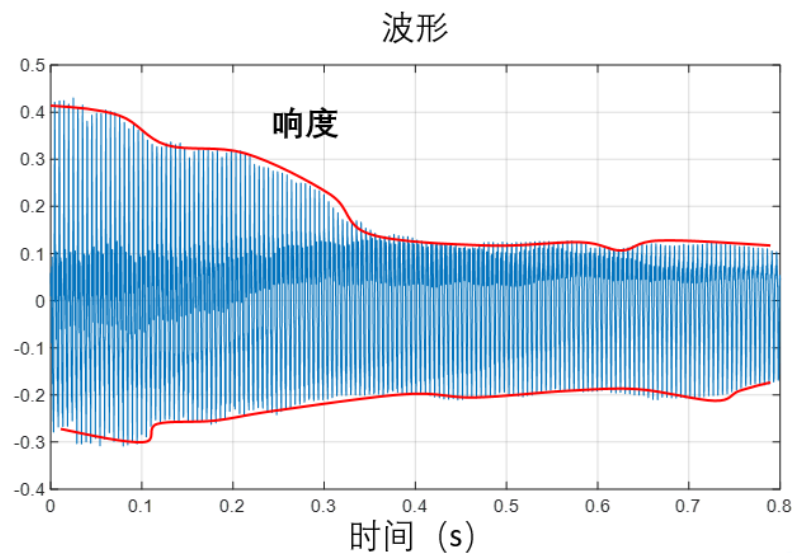
頻譜上的基音頻率反映了音調

頻譜上各個泛音的幅度反映了音色

時域特徵

頻域特徵

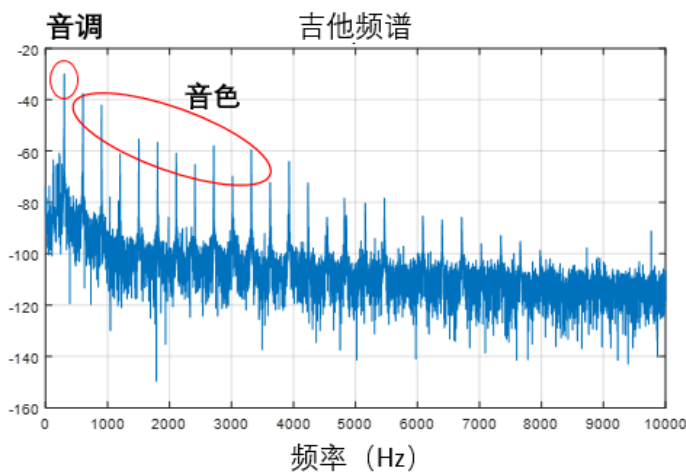
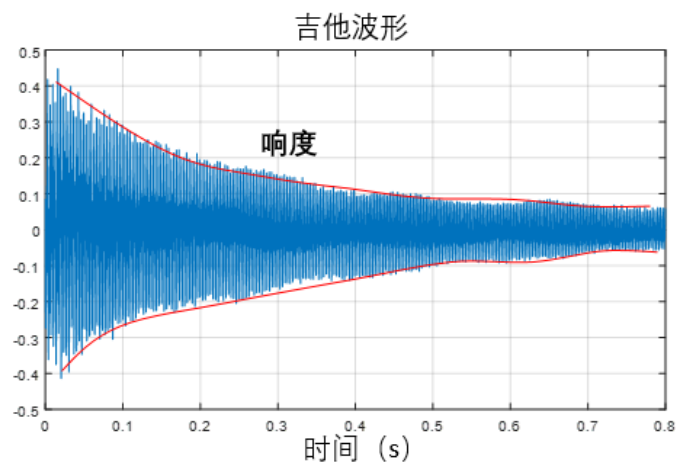
頻域特徵



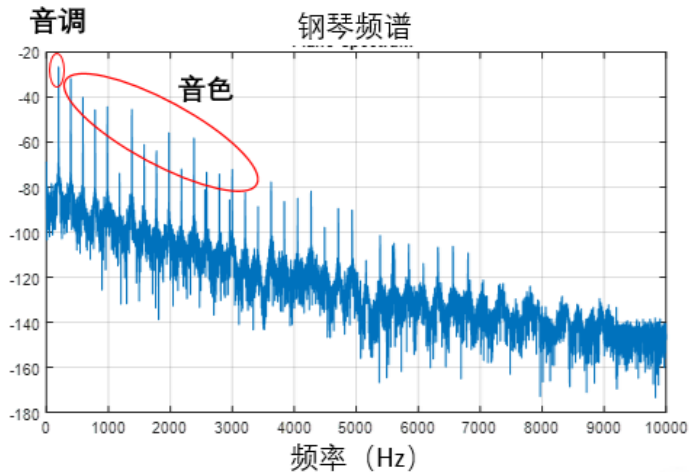
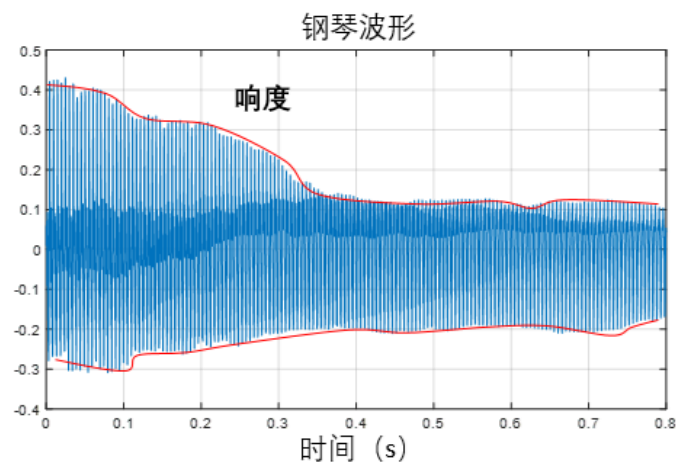
如何從數位化表達中提取聲音的重要特徵



吉他單音的波形
與頻譜



鋼琴單音的波形
與頻譜

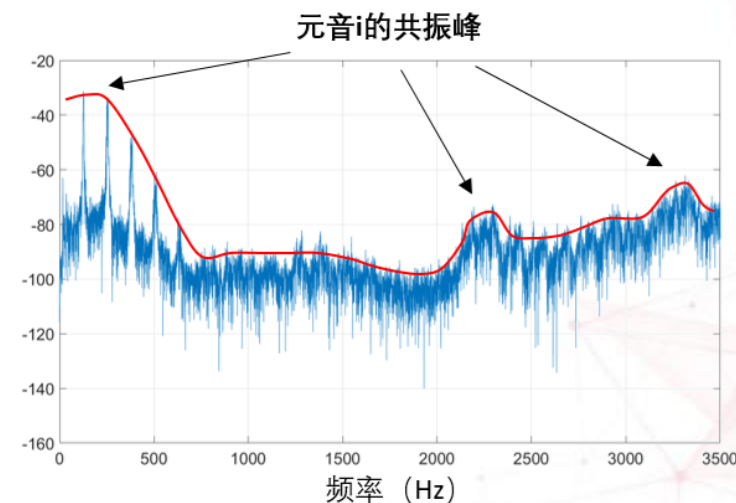
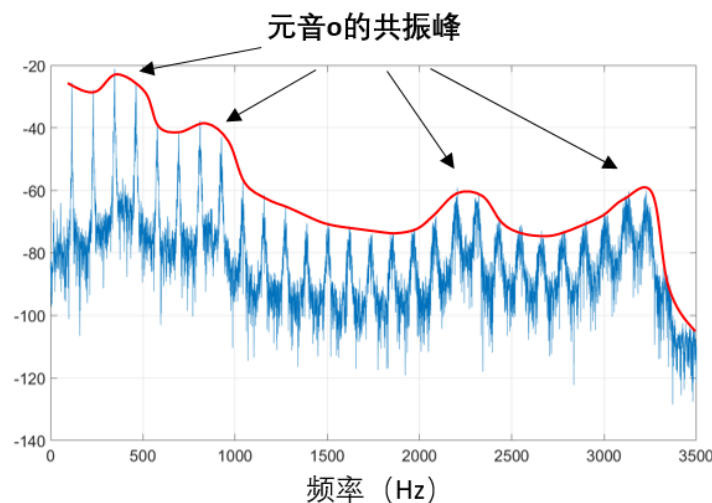


如何從數位化表達中提取聲音的重要特徵

頻譜的形狀刻畫了什麼？

頻譜的包絡形狀可以表達出聲音的一個重要特性——共振峰(formant)。

共振峰(formant) 指的是聲音訊譜上能量相對集中的一些區域。共振峰在語音的分析中較為常用, 因為它在母音(vowel) 的頻譜上十分明顯, 而且不同母音的共振峰也有顯著的區別。



SenseStudy課程平臺 “人工智能基礎 ”： 實驗4 – 2 用频谱图分析乐音的特点

```
piano = data.get('piano')
guitar = data.get('guitar')

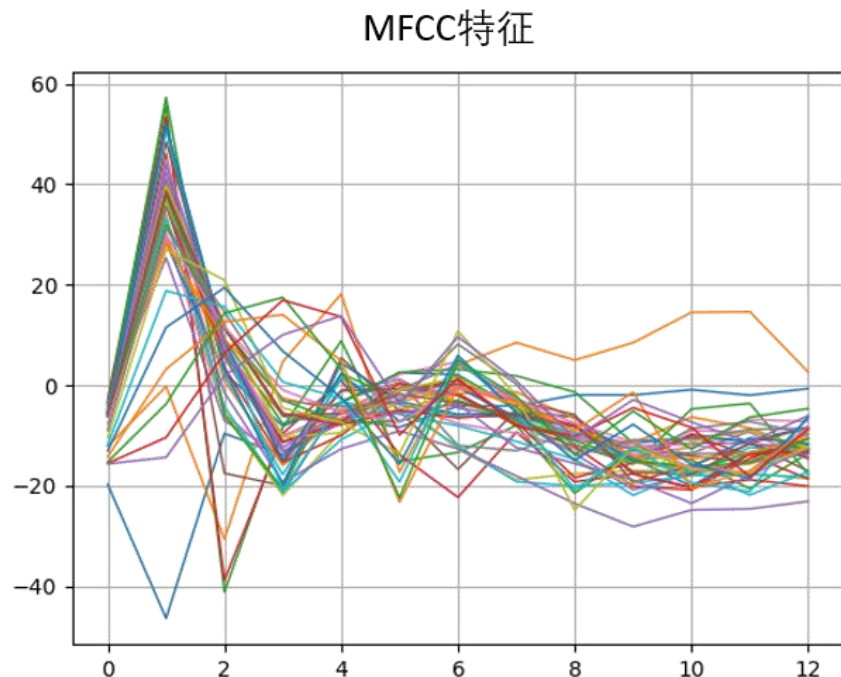
piano_seg = piano.cut(t_start=1.6, t_end=2.4)
guitar_seg = guitar.cut(t_start=1.6, t_end=2.4)

fig(2,1) + [
    plot(piano_seg, type='waveform'),
    plot(guitar_seg, type = 'waveform' )
]

fig(2,1) + [
    plot(piano_seg, type='spectrum'),
    plot(guitar_seg, type = 'spectrum' )
]
```



梅爾頻率倒譜系數(MFCC)

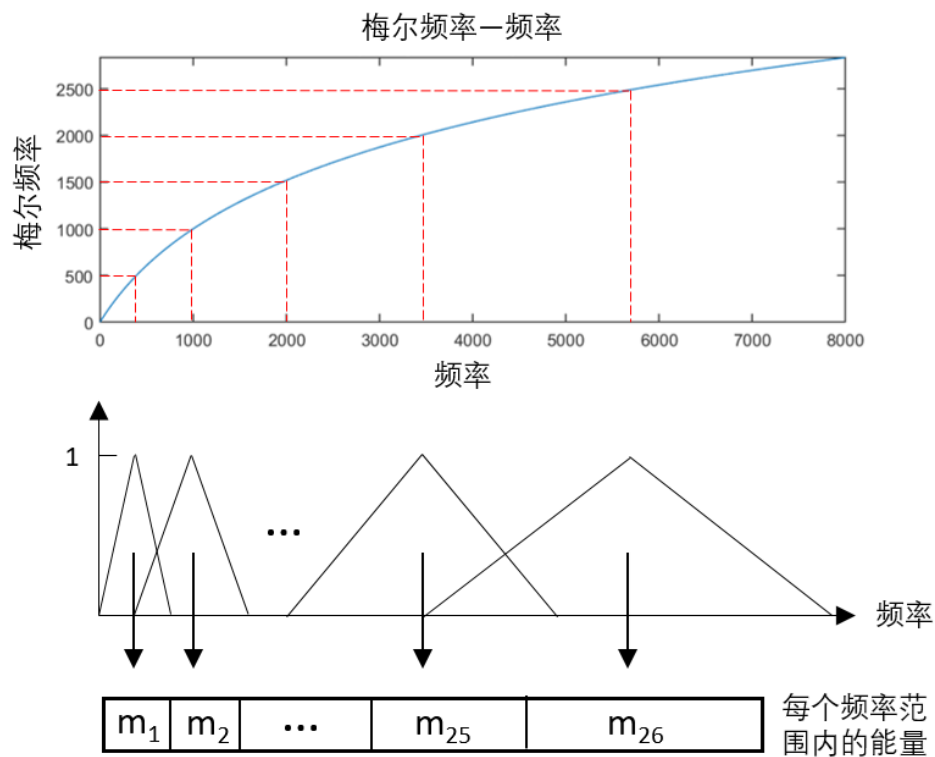


梅爾頻率倒譜系數(MFCC)

梅爾頻率倒譜系數(Mel-Frequency Cepstral Coefficients, MFCC)是一個被廣泛使用的音訊特徵。它的維度比較低，可以粗略地刻畫出頻譜的形狀，因而可以大致描述出不同頻率聲音的能量高低。

利用人耳聽覺對低頻更為敏感的原理，什麼數學方法可以實現“凸顯某一部分、抑制另一部分”的類似功能呢？ 答：對數函數。

梅爾頻率倒譜系數(MFCC)



梅爾頻率(Mel-Frequency)

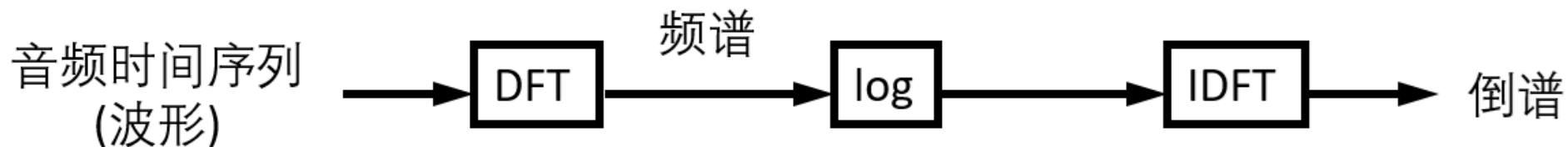
梅爾頻率是一種特殊的頻率刻度, 它與普通頻率的函數關係為

$$\text{mel}(f) = 1125 \times \ln \left(1 + \frac{f}{700} \right)$$

梅爾頻率刻度下等長的頻率區間對應到普通頻率下變為不等長的區間。在低頻部分解析度高, 高頻部分的解析度低。

在每一個頻率區間對頻譜求均值, 它代表了每個頻率範圍內聲音能量的大小。一共有26 個頻率範圍, 從而得到26 維的特徵。

梅爾頻率倒譜系數(MFCC)



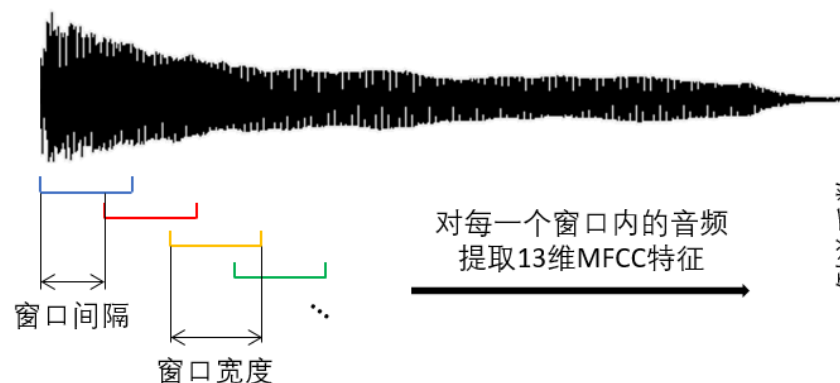
倒譜 (Cepstrum)

在數學上，音訊的時間序列經過離散傅裡葉變換得到頻譜。

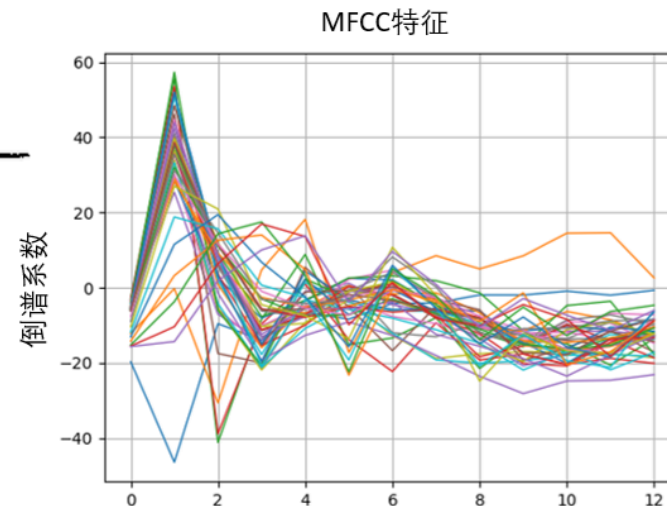
頻譜取對數再經過逆離散傅裡葉變換得到倒譜。

倒譜以至少 $1/n$ 的速度衰減，因而通常取倒譜的前若干係數就可以包含頻譜的大部分資訊。在MFCC特徵裡這個值取13，即把在梅爾頻率下處理過的26維頻譜特徵降低到13維。

梅爾頻率倒譜系數(MFCC)



对每一个窗口内的音频
提取13维MFCC特征



MFCC的提取過程

先把音訊切分為等間隔的若干小段（可重疊），然後對每一小段分別提取13 維的MFCC 特徵。

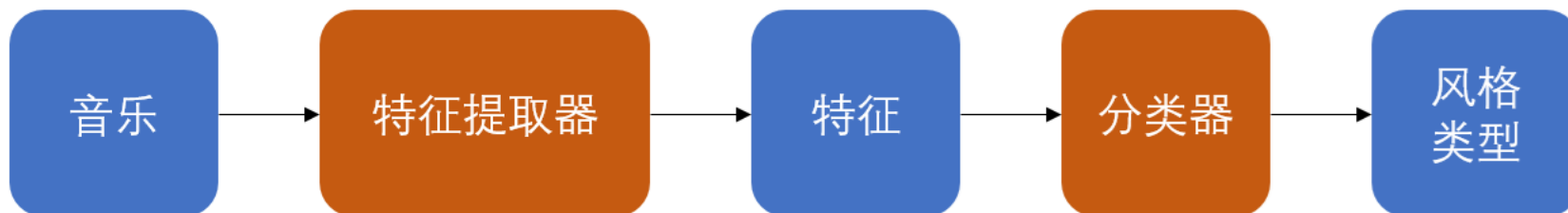
在切分音頻的時候有視窗寬度和視窗間隔兩個參數, 這些參數可以根據音訊的特點進行調節。一種典型的參數是窗口寬度25 毫秒, 窗口間隔10 毫秒。

SenseStudy課程平臺 “人工智能基礎 ”： 實驗4 – 3 观察并理解MFCC特征

```
gtzan = data.get('gtzan')  
  
index = 10  
music, label = gtzan[index]  
  
music_seg1 = music.cut(1.5, 1.53)  
music_seg2 = music.cut(5.5, 5.53)  
  
mfcc = MFCCExtractor()  
  
mfcc_feature1 = mfcc(music_seg1)  
mfcc_feature2 = mfcc(music_seg2)  
  
fig(2,2) + [  
    plot(music_seg1, type='spectrum'),  
    plot(mfcc_feature1, type='mfcc'),  
    plot(music_seg2, type='spectrum'),  
    plot(mfcc_feature2, type='mfcc'),  
]
```



應用實例 -- 音樂風格分類



音樂風格分類流程圖

與圖像分類相似，音樂風格分類的關鍵在於提取特徵。

若要直接使用神經網路完成特徵提取和分類，網路的複雜度會比較高，因為原始的音訊時間序列很長。

因而我們採用預先提取好的MFCC特徵作為輸入，通過卷積神經網路完成特徵的再提煉和分類。

SenseStudy課程平臺 “人工智能基礎 ”： 實驗4 – 4 使用MFCC特征和神经网络完成音乐风格分类

```
dataset = data.get('gtzan-mfcc')

label_names = dataset.meta['label_names']
print(label_names)

trainset, testset = dataset.split(8,2)

net = CNNClassifier(backbone = AudioNet(), in_shape=(1, 624, 13), num_classes=10)

net.train(trainset, alg=SGD(lr=0.001, epoch=5, bs=16))

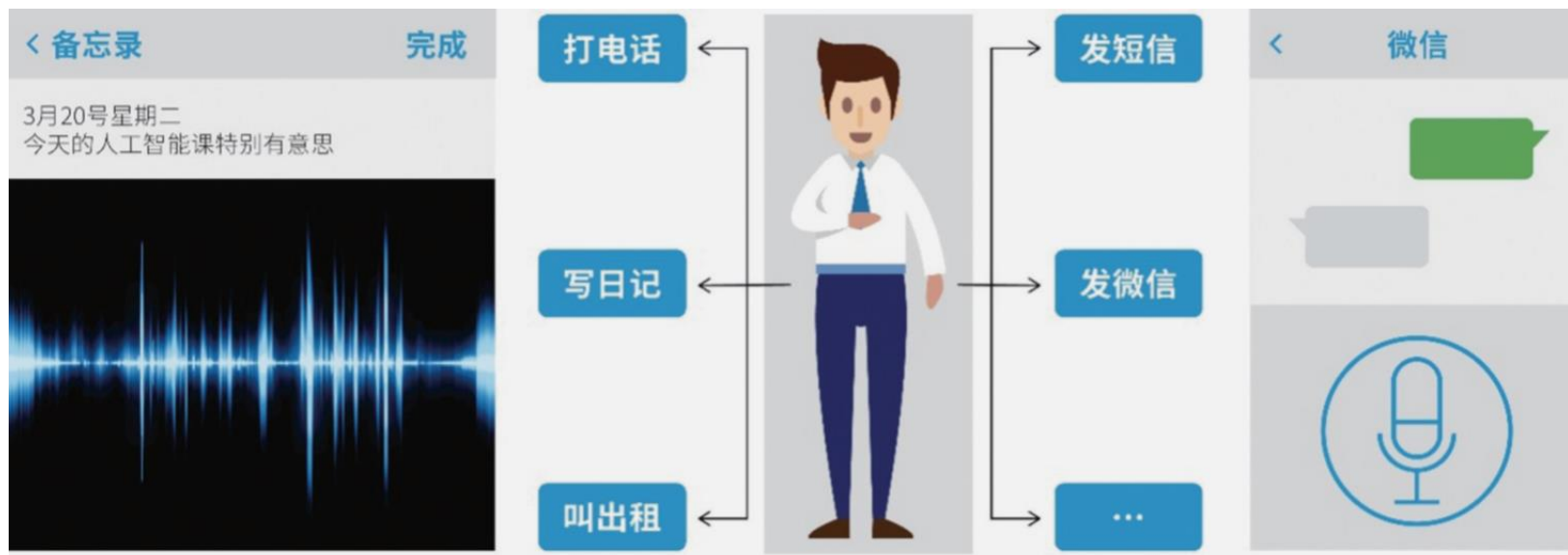
acc = net.accuracy(testset)
print('Accuracy: ', acc)

index = 10
mfcc_feature, label = testset[index]
label_name = label_names[label]

print('Ground Truth: ', label_name)

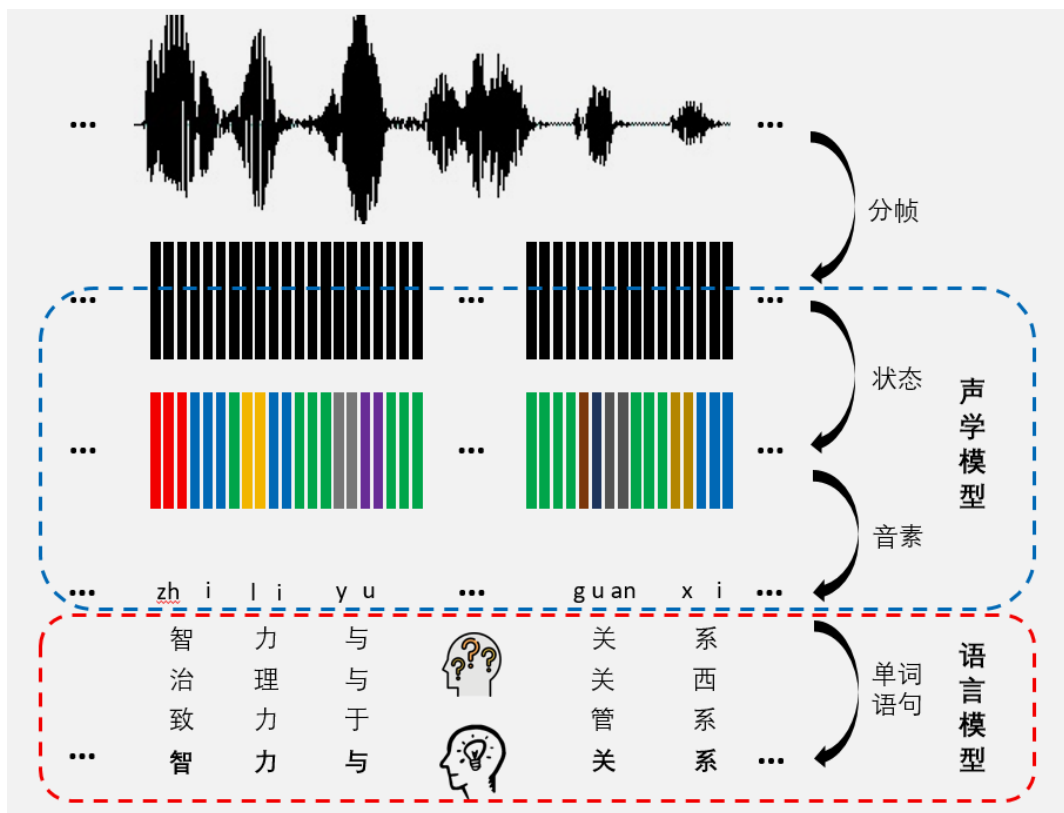
pred = net.predict(mfcc_feature)
pred_name = label_names[pred]
print('Prediction: ', pred_name)
```





語音辨識

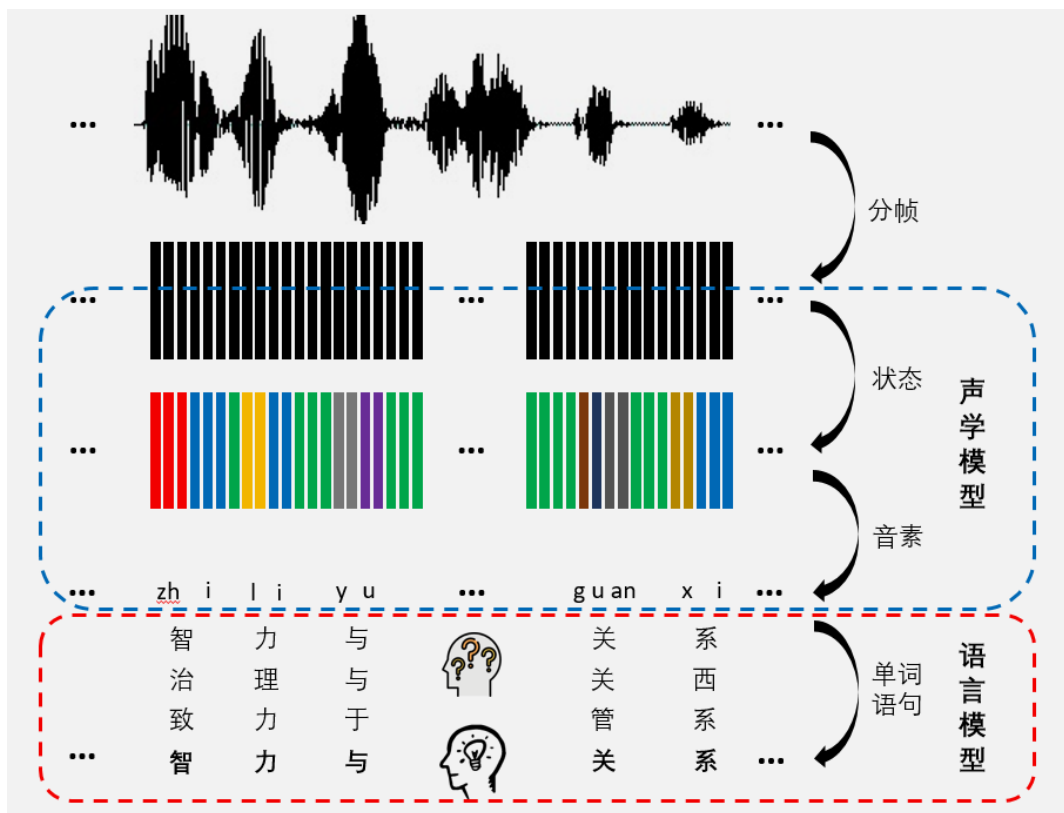
語音辨識(speech recognition) 的目的是把人說的話轉化為文字或者機器可以理解的指令，從而實現人與機器的語音交流。語音辨識技術已經在現實生活中得到了廣泛的應用。



語音辨識原理

首先把一段語音分成若干小段, 這個過程稱為分幀。然後把每一幀識別為一個狀態, 再把狀態組合成音素, 最後組成意義明確的語句。

音素一般就是我們熟知的聲母和韻母, 而狀態則是比音素更加細節的語音單位, 一個音素通常會包含三個狀態。



語音辨識原理

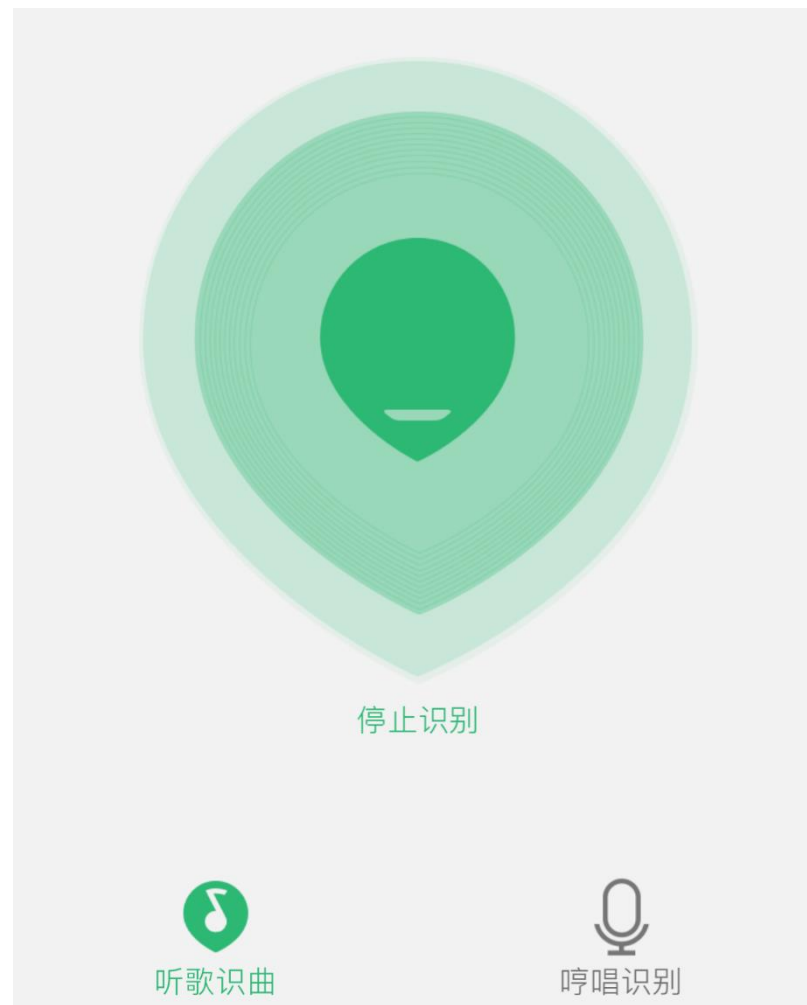
把一系列語音幀轉換為若干音素的過程利用了語言的聲學特性, 因而這一部分被稱為聲學模型(acoustic model)。

從音素到文字的過程需要用到語言表達的特點, 這樣才能從同音字中挑選出正確的文字, 組成意義明確的語句, 這部分被稱為語言模型(language model)。

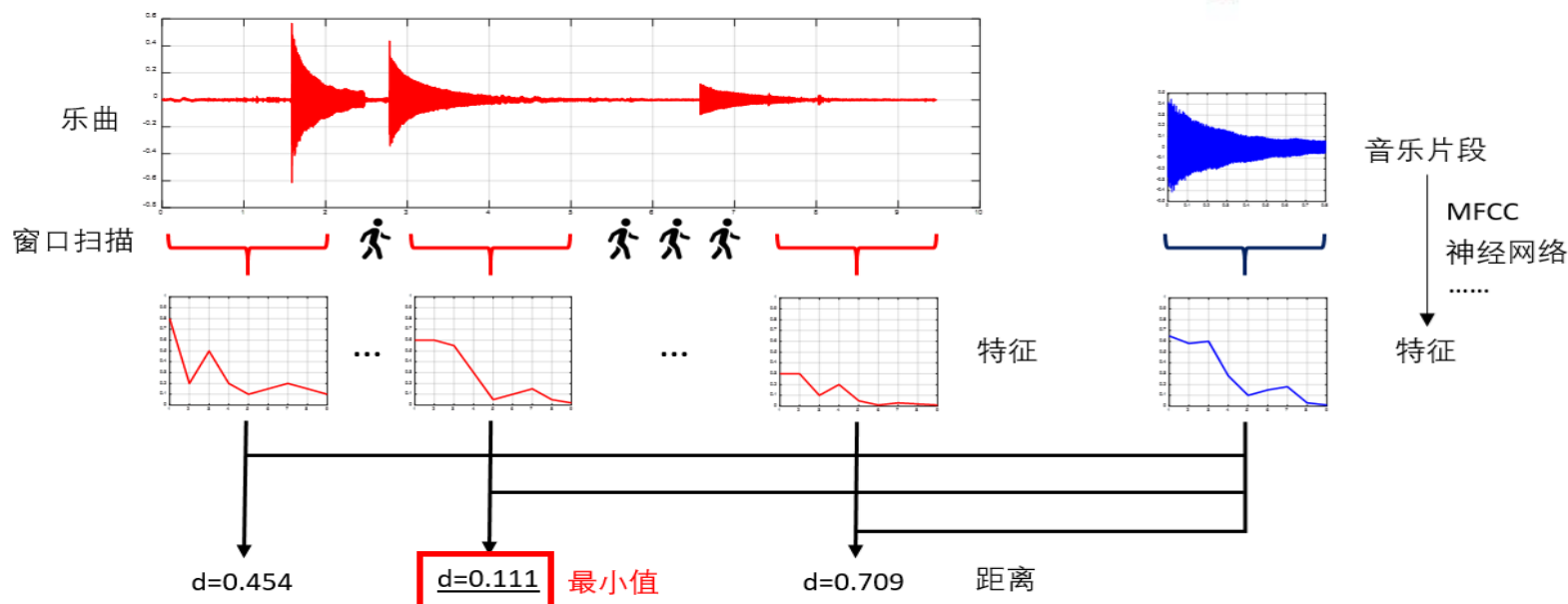
樂曲檢索

樂曲檢索任務的輸入是一個很短的音樂片段, 而輸出是資料庫中與輸入片段最為相似的樂曲。

在一些音樂應用中有一個有趣的功能, 根據使用者哼唱的一個片段找出相對應的歌曲, 這是樂曲檢索的一個典型應用。



應用實例 -- 樂曲檢索

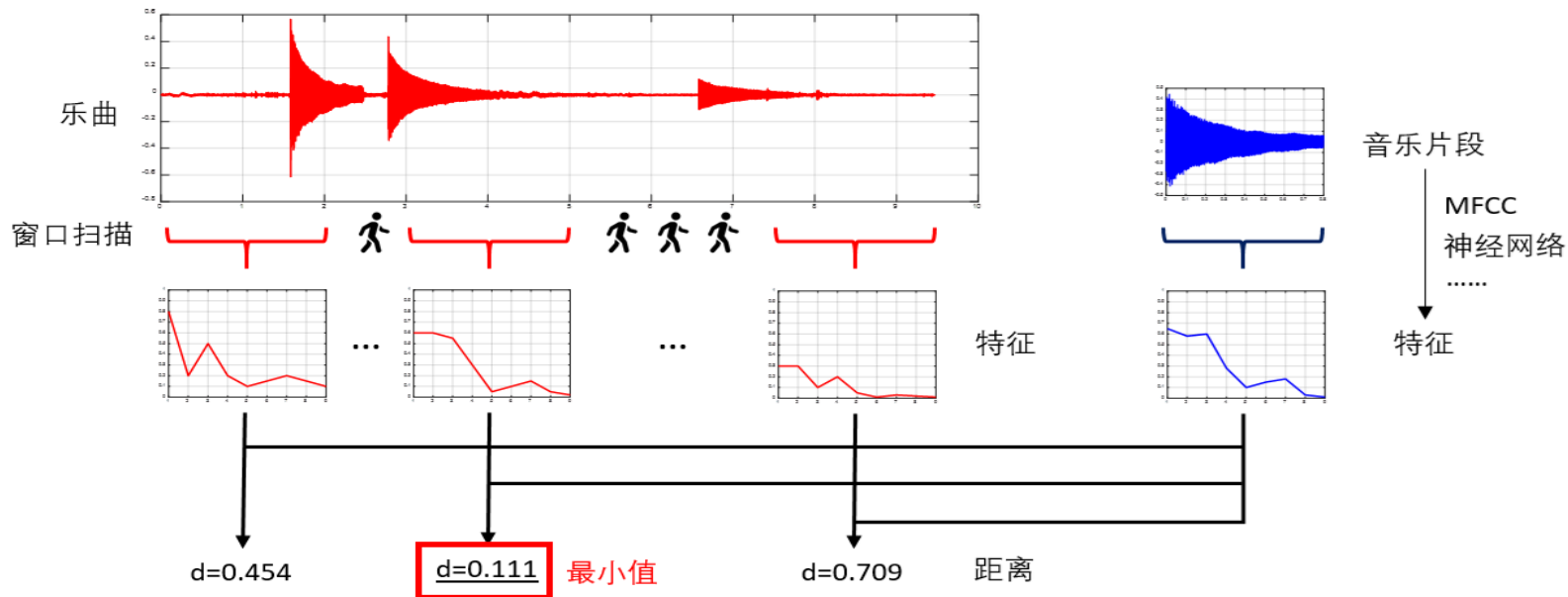


樂曲檢索原理

樂曲檢索與文檔中的精確查找不同, 這裡的查找是模糊的, 比如可能是不同歌手演唱的同一曲目, 雖然相似但不全相同。所以我們並不能直接評判“找到”和“沒找到”, 而應該給出一個相似度。

我們通常用一個距離來度量相似度, 如歐式距離, 給定兩組特徵 $x = (x_1, x_2, x_3)$, $y = (y_1, y_2, y_3)$ 它們的距離是

$$d = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2}$$



樂曲檢索原理

在樂曲上按照時間順序依次截取和音樂片段長度一致的段落, 相鄰段落之間的時間間隔可大可小, 通常要保證它們在時間上有較大的重疊, 這一過程被稱為“視窗掃描”。

然後計算片段和所截段落的特徵並算出它們的距離, 取這些距離的最小值作為音樂片段與樂曲的距離。最終與音樂片段距離最小的樂曲即為檢索的結果。