**目錄**

[01 系統需求 1](#_Toc12303201)

[02 資料集 2](#_Toc12303202)

[A. 歌手資料集 2](#_Toc12303203)

[B. 專輯資料集 2](#_Toc12303204)

[C. 歌曲特徵資料集 2](#_Toc12303205)

[D. 推薦歌曲資料集 2](#_Toc12303206)

[E. 調查歌曲資料集 2](#_Toc12303207)

[F. 資料缺失 2](#_Toc12303208)

[03 音頻下載 3](#_Toc12303209)

[A. 預備步驟 3](#_Toc12303210)

[B. 執行步驟 3](#_Toc12303211)

[04 音頻嵌入 4](#_Toc12303212)

[A. 執行ipynb檔 4](#_Toc12303213)

[B. 修改路徑 4](#_Toc12303214)

[C. 預訓練模型pre-trained 5](#_Toc12303215)

[D. 特徵向量 5](#_Toc12303216)

[05 相似性計算 6](#_Toc12303217)

**WaveNet Embedding Document**

# 系統需求

* magenta
* numpy
* librosa
* tensorflow
* 安裝CUDA和cuDNN

GPU版的TensorFlow需多加裝CUDA和cuDNN，先確認顯卡是否支援CUDA。

確認安裝環境：顯卡版本要先檢查

本次版本：tensorflow-gpu-1.12.0、CUDA：v9.0 (cuDNN：v7.3.1) for window10 64bit

<https://medium.com/@rick.huang1609/window-10%E5%AE%89%E8%A3%9Dtensorflow-gpu%E4%B8%A6%E5%9C%A8jupyter-notebook%E5%92%8Cspyder%E9%81%8B%E8%A1%8C-221bb4707dbd>

# 資料集

* 1. 歌手資料集

|  |  |  |  |
| --- | --- | --- | --- |
| tracks\_artist.csv | | | |
| id | name | genres | img\_url |
| 歌手ID | 歌手名稱 | 類型 | 歌手圖片 |
| https://open.spotify.com/artist/5Z1CCuBsyhEHngq3U5IraY | | | |

* 1. 專輯資料集

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| tracks\_album.csv | | | | | | |
| id | name | preview\_url | img\_url | tracks\_number | release\_date | artist\_id |
| 專輯ID | 專輯名稱 | 專輯網址 | 專輯圖片 | 專輯歌曲數量 | 發行日期 | 歌手ID |
| https://open.spotify.com/album/22pfCAdDOgyC2JfSi5OwxT | | | | | | |

* 1. 歌曲特徵資料集

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| tracks\_features.csv | | | | | | |
| id | name | preview\_url | img\_url | tracks\_number | release\_date | artist\_id |
| 專輯ID | 專輯名稱 | 專輯網址 | 專輯圖片 | 專輯歌曲數量 | 發行日期 | 歌手ID |

* 1. 推薦歌曲資料集

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| trakcs\_data\_for\_rec.csv | | | | | | | |
| name | id | artist\_name | track\_  preview\_url | artist\_  img\_url | album\_name | album\_id | img\_url |
| 歌曲  名稱 | 歌手ID | 歌手名稱 | 歌曲網址 | 歌手  圖片 | 專輯名稱 | 專輯ID | 專輯圖片 |

* 1. 調查歌曲資料集

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| name | id | artist\_name | artist\_id | preview\_url |
| 歌曲名稱 | 歌曲ID | 歌手名稱 | 歌手ID | 歌曲網址 |

* 1. 音頻歌詞向量資料集

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| .csv | | | | | | | |
| id | name | audio\_  vector | audio\_  top20 | audio\_  euc | lyrics\_  vector | lyrics \_top20 | lyrics \_  euc | |
| 歌曲ID | 歌曲名稱 | 音頻向量 | 音頻前20 | 音頻距離 | 音頻向量 | 音頻前20 | 音頻距離 | |

* 1. 資料缺失
* track\_album.csv

專輯ID 52O1oc2EC6kwNo8iwBs919 → 沒有專輯圖片

專輯ID 70HXQp5RAjM37JHBuvL8Dg → 沒有專輯圖片

* trakcs\_data\_for\_rec.csv

專輯ID 70HXQp5RAjM37JHBuvL8Dg → 沒有專輯圖片

* tracks\_data\_for\_rec.csv

專輯ID 70HXQp5RAjM37JHBuvL8Dg → 沒有專輯圖片

# 音頻下載

* 1. 預備步驟

音頻網址放入csv以供讀取

* 1. 執行步驟
     1. 執行ipynb檔案

執行audio\_download.ipynb

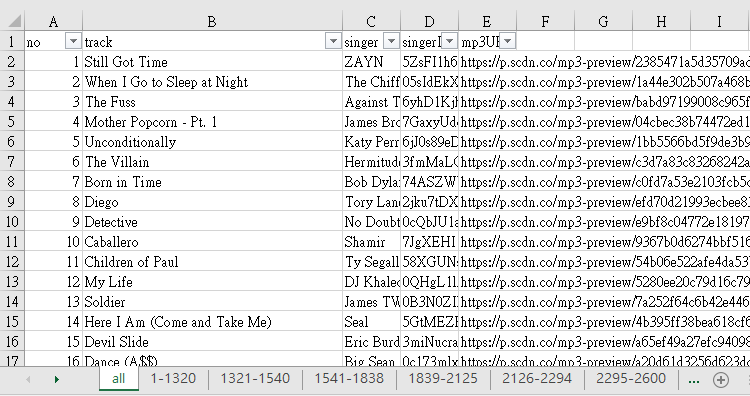
* + 1. 修改路徑

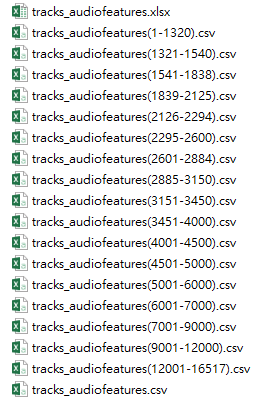
audio = pd.read\_csv('./tracks\_audiofeatures/tracks\_audiofeatures(9001-12000).csv')

* + 1. 檔名命名

依照trakcs\_data\_for\_rec.csv之順序編號

參照檔案tracks\_audiofeatures.xlsx

D:\Anaconda3\Scripts\7 實驗進度\wavenet\_embedding-master\tracks\_audiofeatures

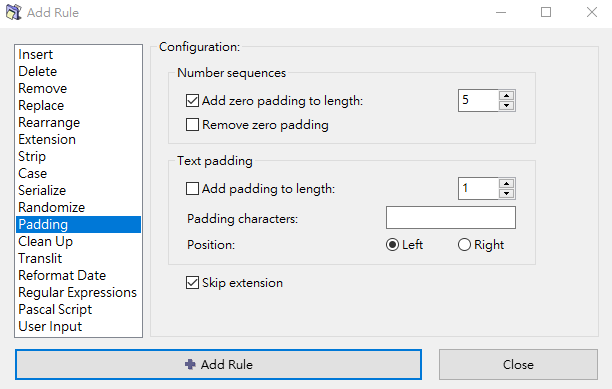
* + 1. 完成下載

音頻下載於設定之資料夾tracks\_audiofeatures

* + 1. 後製動作

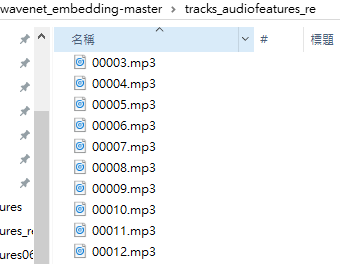
使用ReNamer工具 (D:\renamer-6.7) 標準化命名

→ 選擇padding補0，長度為5



標準化命名結果：

資料夾：wavenet\_embedding-master\tracks\_audiofeatures\_re



* 1. 音頻修改
  + 1492
  + 3264

# 音頻嵌入

* 1. 執行ipynb檔

wavenet\_embedding\_vector-Copy1

* 1. 修改路徑

dataset\_path = './tracks\_audiofeatures0625\_re' #dataset位置

with open('tracks\_audiofeatures0625\_re/audio\_vector.csv', 'a', newline='') as csvFile:

data = csv.reader(open('tracks\_audiofeatures0625\_re/audio\_vector.csv'))

with open("tracks\_audiofeatures0625\_re/audio\_vector\_sort.csv", "w") as f:

* 1. 預訓練模型pre-trained

呼叫預訓練模型 def wavenet\_encode(wave):

* + 1. 下載WaveNet模型 (wavenet embedding pretrain model)

The Models : WaveNet

<https://github.com/tensorflow/magenta/tree/master/magenta/models/nsynth>

<http://download.magenta.tensorflow.org/models/nsynth/wavenet-ckpt.tar>

* + 1. 存放於專案底下

model\_path = './wavenet-ckpt/wavenet-ckpt/model.ckpt-200000' #模型位置

* 1. 特徵向量

呼叫特徵提取函式　def wavenet\_vector(dataset\_path):

1. 判斷音頻是否已嵌入過

比較mp3的名稱和路徑是否重複，若重覆則不跑後面特徵提取步驟。

comparison\_paths = list(set(mp3\_paths).difference(set(replace\_paths)))

comparison\_names = list(set(mp3\_names).difference(set(replace\_names)))

1. 特徵向量提取

* 修改CSV路徑

with open('tracks\_audiofeatures0625\_re/audio\_vector.csv', 'a', newline='') as csvFile:

* 分成16聲道，各取3個特徵

std\_wavenet = np.std(wavenet\_data, axis=0) # 標準差

mean\_wavenet = np.mean(wavenet\_data, axis=0) # 平均數

average\_difference\_channels = np.zeros((16,)) # 平均差

average\_difference\_channels = np.array(average\_difference\_channels)

* 向量連接

concat\_features\_wavenet = np.hstack((std\_wavenet, mean\_wavenet))

concat\_features\_wavenet = np.hstack((concat\_features\_wavenet, average\_difference\_channels))

1. 存取.npy檔

concat\_features\_wavenet：每個音頻的特徵向量結果

save\_path = mp3.replace('.mp3', "")

string = save\_path

num = string[-5:]

print('save',concat\_features\_wavenet.shape,'in',save\_path)

np.save(save\_path,concat\_features\_wavenet) # 存特徵向量在原資料夾

1. 寫入CSV

添加新列至CSV，使用'a'

with open('路徑/audio\_vector.csv', 'a', newline='') as csvFile:

writer = csv.writer(csvFile)

Table = [[num,concat\_features\_wavenet]]

writer.writerows(Table)

1. 排序歌曲向量CSV

* 使用python operator模組
* 運用operator.itemgetter取出想排序的元素

import operator

data = csv.reader(open('tracks\_audiofeatures0625\_re/audio\_vector.csv'))

sortedlist = sorted(data, key=operator.itemgetter(0))

operator.itemgetter(0)代表利用第0個column來作為排序依據。

【補充】

sorted函數也可以進行多級排序，例如要根據第2個域和第3個域進行排序，

sorted(data, key=operator.itemgetter(1,2))。

呼叫函式sorted時，以參數key傳入lambda，指定要根據哪個元素進行排序。

根據age年齡排序：print(sorted(data, key=lambda x: x['age']))

先排age年齡、再排name名字：print(sorted(data, key=lambda x: (x['age'],['name'])))

<http://yehnan.blogspot.com/2015/06/pythonoperatoritemgetter.html>

* 修改路徑

with open("tracks\_audiofeatures0625\_re/audio\_vector\_sort.csv", "w") as f:

fileWriter = csv.writer(f)

fileWriter.writerow(['num','audio\_vector']) # 欄位標題

for row in sortedlist:

fileWriter.writerow(row)

* 刪除空白列

<https://kknews.cc/zh-tw/tech/l8v33m9.html>

<https://isvincent.pixnet.net/blog/post/43204702-excel-%E5%88%AA%E9%99%A4%E5%9B%BA%E5%AE%9A%E9%96%93%E9%9A%94%E5%88%97>

# 相似性計算

執行cos\_euc\_output.py