

原

音频处理库—librosa的安装与使用

2018年03月18日 23:00:58

z小白

阅读数 20816

更多

 版权声明：本文为博主原创文章，请尊重原创，转载请注明原文地址和作者信息！ <https://blog.csdn.net/zzc15806/article/details/79603994>

目录

序言

一、librosa安装

- pypi
- conda
- source

二、librosa常用功能

- 核心音频处理函数
- 音频处理
- 频谱表示
- 幅度转换
- 时频转换
- 特征提取
- 绘图显示

三、常用功能代码实现

- 读取音频
- 提取特征
- 提取Log-Mel Spectrogram 特征
- 提取MFCC特征
- 绘图显示
- 绘制声音波形
- 绘制频谱图

序言

Librosa是一个用于音频、音乐分析、处理的python工具包，一些常见的时频处理、特征提取、绘制声音图形等功能应有尽有，功能十分强大。的
安装与使用方法。

一、librosa安装

Librosa官网提供了多种安装方法，详细如下：

pipi

最简单的方法就是进行pip安装，可以满足所有的依赖关系，命令如下：

```
pip install librosa
```

conda

如果安装了Anaconda，可以通过conda命令安装：

```
conda install -c conda-forge librosa
```

source

直接使用源码安装，需要提前下载源码（<https://github.com/librosa/librosa/releases/>），通过下面命令安装：

```
1 | tar xzf librosa-VERSION.tar.gz
2 | cd librosa-VERSION/
3 | python setup.py install
```

二、librosa常用功能

核心音频处理函数

这部分介绍了最常用的音频处理函数，包括音频读取函数load()，重采样函数resample()，短时傅里叶变换stft()，幅度转换函数amplitude_to_db()函数hz_to_mel()等。这部分函数很多，详细可参考librosa官网 <http://librosa.github.io/librosa/core.html>

音频处理

load (path[, sr, mono, offset, duration, ...])	Load an audio file as a floating point time series.
to_mono [y]	Force an audio signal down to mono.
resample [y, orig_sr, target_sr, res_type, ...]	Resample a time series from orig_sr to target_sr
get_duration ([y, sr, S, n_fft, hop_length, ...])	Compute the duration (in seconds) of an audio time series.
autocorrelate [y[, max_size, axis]]	Bounded auto-correlation
zero_crossings [y[, threshold, ...]]	Find the zero-crossings of a signal y: indices i such that sign
concat ([times, frames, sr, hop_length, ...])	Returns a signal with the signal 'times' placed at each spec

频谱表示

stft [y[, n_fft, hop_length, win_length, ...]]	Short-time Fourier transform (STFT)
istft (stft_matrix[, hop_length, win_length, ...])	Inverse short-time Fourier transform (ISTFT).
hgram [y[, sr, n_fft, hop_length, ...]]	Compute the instantaneous frequency (as a proportion
cqt [y[, sr, hop_length, fmin, n_bins, ...]]	Compute the constant-Q transform of an audio signal.
icqt [C[, sr, hop_length, fmin, ...]]	Compute the inverse constant-Q transform.
hybrid_cqt [y[, sr, hop_length, fmin, ...]]	Compute the hybrid constant-Q transform of an audio
pseudo_cqt [y[, sr, hop_length, fmin, ...]]	Compute the pseudo constant-Q transform of an audio
lfrf [y[, sr, win_length, hop_length, ...]]	Time-frequency representation using ER filters (ERF).
mf [y[, f_min, n_fft, kind, beta, ...]]	The fast Melin transform (FMT) [R1312] of a uniformly
extract_harmonics [x, freqs, h_range[, kind, ...]]	Compute the energy at harmonics of time-frequency re
salience [S, freqs, h_range[, weights, ...]]	Harmonic salience function.
phase_vocoder [D, rate[, hop_length]]	Phase vocoder.
magnitude [D[, power]]	Separate a complex-valued spectrogram D into its mag

幅度转换

<code>amplitude_to_db(S[, ref, amin, top_db])</code>	Convert an amplitude spectrogram to dB-scaled spectrogram
<code>db_to_amplitude(S, db[, ref])</code>	Convert a dB-scaled spectrogram to an amplitude spectrogram
<code>power_to_db(S[, ref, amin, top_db])</code>	Convert a power spectrogram (amplitude squared) to dB-scaled spectrogram
<code>db_to_power(S, db[, ref])</code>	Convert a dB-scale spectrogram to a power spectrogram
<code>perceptual_weighting(S, frequencies, "twangs")</code>	Perceptual weighting of a power spectrogram
<code>A_weighting(frequencies[, min_db])</code>	Compute the A-weighting of a set of frequencies. ⁽¹⁾

时频转换

<code>frames_to_samples(frames[, hop_length, n_fft])</code>	Converts frame indices to audio sample indices
<code>frames_to_time(frames[, sr, hop_length, n_fft])</code>	Converts frame counts to time (seconds)
<code>samples_to_frames(samples[, hop_length, n_fft])</code>	Converts sample indices into STFT frames.
<code>samples_to_time(samples[, sr])</code>	Convert sample indices to time (in seconds).
<code>time_to_frames(times[, sr, hop_length, n_fft])</code>	Converts time stamps into STFT frames.
<code>time_to_samples(times[, sr])</code>	Convert timestamps (in seconds) to sample indices.
<code>hz_to_note(frequencies, "twangs")</code>	Convert one or more frequencies (in Hz) to the nearest MIDI note number(s)
<code>hz_to_note(frequencies)</code>	Get MIDI note number(s) for given frequencies
<code>note_to_hz(note)</code>	Get the frequency (Hz) of MIDI note(s)
<code>note_to_note(midi[, octave, cents])</code>	Convert one or more MIDI numbers to note strings.
<code>note_to_hz(note, "twangs")</code>	Convert one or more note names to frequency (Hz)
<code>note_to_note(note[, round_midi])</code>	Convert one or more spelled notes to MIDI number(s).
<code>hz_to_mel(frequencies[, Hz])</code>	Convert Hz to Mels
<code>hz_to_octa(frequencies[, A440])</code>	Convert frequencies (Hz) to (fractional) octave number
<code>mel_to_hz(mels[, Hz])</code>	Convert melbin numbers to frequencies
<code>octa_to_hz(octa[, A440])</code>	Convert octaves numbers to frequencies.
<code>fft_frequencies([sr, n_fft])</code>	Alternative implementation of <code>np.fft.ffreqs</code>
<code>cent_frequencies([n_bins, fmin[, ...]])</code>	Compute the center frequencies of Constant-Q bins.

特征提取

本部分列举了一些常用的频谱特征的提取方法，包括常见的Mel Spectrogram、MFCC、CQT等。函数详细信息可参考<http://librosa.github.io/librosa/feature.html>

<code>chroma_mfcc</code> (<code>y</code> , <code>M</code> , <code>S</code> , <code>n_mfcc</code> , <code>n_mels</code> , ...)	Compute a chromagram from a waveform or power spectrogram.
<code>chroma_mel</code> (<code>y</code> , <code>M</code> , <code>C</code> , <code>hop_length</code> , <code>freq</code> , ...)	Constant-Q chromagram
<code>chroma_mse</code> (<code>y</code> , <code>M</code> , <code>C</code> , <code>hop_length</code> , <code>freq</code> , ...)	Computes the chroma variant "Chroma (Energy Normalized)" (CENs), following [R3131].
<code>mel_spectrogram</code> (<code>y</code> , <code>M</code> , <code>S</code> , <code>n_mels</code> , ...)	Compute a mel-scaled spectrogram.
<code>mfcc</code> (<code>y</code> , <code>M</code> , <code>S</code> , <code>n_mfcc</code>)	Mel-frequency cepstral coefficients
<code>rms</code> (<code>y</code> , <code>S</code> , <code>frame_length</code> , <code>hop_length</code> , ...)	Compute root-mean-square (RMS) energy for each frame, either from the audio samples <code>y</code> or from a spectrogram <code>S</code> .
<code>spectral_centroid</code> (<code>y</code> , <code>M</code> , <code>S</code> , <code>n_mels</code> , ...)	Compute the spectral centroid.
<code>spectral_bandwidth</code> (<code>y</code> , <code>M</code> , <code>S</code> , <code>n_mels</code> , ...)	Compute <code>p</code> th-order spectral bandwidth.
<code>spectral_contrast</code> (<code>y</code> , <code>M</code> , <code>S</code> , <code>n_mels</code> , ...)	Compute spectral contrast [R3333].
<code>spectral_flatness</code> (<code>y</code> , <code>S</code> , <code>n_mels</code> , <code>hop_length</code> , ...)	Compute spectral flatness
<code>spectral_rolloff</code> (<code>y</code> , <code>M</code> , <code>S</code> , <code>n_mels</code> , ...)	Compute roll-off frequency
<code>poly_features</code> (<code>y</code> , <code>M</code> , <code>S</code> , <code>n_mels</code> , <code>hop_length</code> , ...)	Get coefficients of fitting an <code>n</code> th-order polynomial to the columns of a spectrogram.
<code>tonnetz</code> (<code>y</code> , <code>M</code> , <code>chroma</code>)	Computes the tonal centroid features (tonnetz), following the method of [R3737].
<code>zero_crossing_rate</code> (<code>y</code> , <code>frame_length</code> , ...)	Compute the zero-crossing rate of an audio time series.

绘图显示

包含了常用的频谱显示函数`specshow()`，波形显示函数`waveplot()`，详细信息请参考<http://librosa.github.io/librosa/display.html>

<code>specshow</code> (<code>data</code> , <code>x_coords</code> , <code>y_coords</code> , <code>x_axis</code> , ...)	Display a spectrogram/chromagram/eq/etc.
<code>waveplot</code> (<code>y</code> , <code>M</code> , <code>max_points</code> , <code>x_axis</code> , ...)	Plot the amplitude envelope of a waveform.
<code>map</code> (<code>data</code> , <code>colmap</code> , <code>cmap_seq</code> , <code>cmap_bool</code> , ...)	Get a default colormap from the given data.
<code>timeformatter</code> (<code>log</code>)	A tick formatter for time axes.
<code>timeformatter</code> (<code>octave</code> , <code>major</code>)	Tick formatter for Notes
<code>logfreqformatter</code> (<code>major</code>)	Tick formatter for logarithmic frequency
<code>chromaformatter</code>	A formatter for chroma axes
<code>tonnetzformatter</code>	A formatter for tonnetz axes (on, off, etc.) [R3806]

三、常用功能代码实现

读取音频

```
1 >>> import librosa
2 >>> # Load a wav file
3 >>> y, sr = librosa.load('./beat.wav')
4 >>> y
5 array([ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00, ...,
6         8.12290182e-06,  1.34394732e-05,  0.00000000e+00], dtype=float32)
7 >>> sr
8 22050
```

Librosa默认的采样率是22050，如果需要读取原始采样率，需要设定参数`sr=None`:

```
1 >>> import librosa
2 >>> # Load a wav file
3 >>> y, sr = librosa.load('./beat.wav', sr=None)
4 >>> sr
5 44100
```

可见, 'beat.wav'的原始采样率为44100。如果需要重采样, 只需要将采样率参数sr设定为你需要的值:

```
1 >>> import librosa
2 >>> # Load a wav file
3 >>> y, sr = librosa.load('./beat.wav', sr=16000)
4 >>> sr
5 16000
```

提取特征

提取Log-Mel Spectrogram 特征

Log-Mel Spectrogram特征是目前在语音识别和环境声音识别中很常用的一个特征, 由于CNN在处理图像上展现了强大的能力, 使得音频信号的频谱愈加广泛, 甚至比MFCC使用的更多。在librosa中, Log-Mel Spectrogram特征的提取只需几行代码:

```
1 >>> import librosa
2 >>> # Load a wav file
3 >>> y, sr = librosa.load('./beat.wav', sr=None)
4 >>> # extract mel spectrogram feature
5 >>> melspec = librosa.feature.melspectrogram(y, sr, n_fft=1024, hop_length=512, n_mels=128)
6 >>> # convert to log scale
7 >>> logmelspec = librosa.power_to_db(melspec)
8 >>> logmelspec.shape
9 (128, 194)
```

可见, Log-Mel Spectrogram特征是二维数组的形式, 128表示Mel频率的维度(频域), 194为时间帧长度(时域), 所以Log-Mel Spectrogram特征的时频表示特征。其中, n_fft指的是窗的大小, 这里为1024; hop_length表示相邻窗之间的距离, 这里为512, 也就是相邻窗之间有50%的overlap mel bands的数量, 这里设为128。

提取MFCC特征

MFCC特征是一种在自动语音识别和说话人识别中广泛使用的特征。关于MFCC特征的详细信息, 有兴趣的可以参考博客<http://blog.csdn.net/zzc15806/article/details/79246716>。在librosa中, 提取MFCC特征只需要一个函数:

```
1 >>> import librosa
2 >>> # Load a wav file
3 >>> y, sr = librosa.load('./beat.wav', sr=None)
4 >>> # extract mfcc feature
5 >>> mfccs = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=40)
6 >>> mfccs.shape
7 (40, 194)
```

关于mfcc, 这里就不在赘述。

Librosa还有很多其他音频特征的提取方法, 比如CQT特征、chroma特征等, 在第二部分“librosa常用功能”给了详细的介绍。

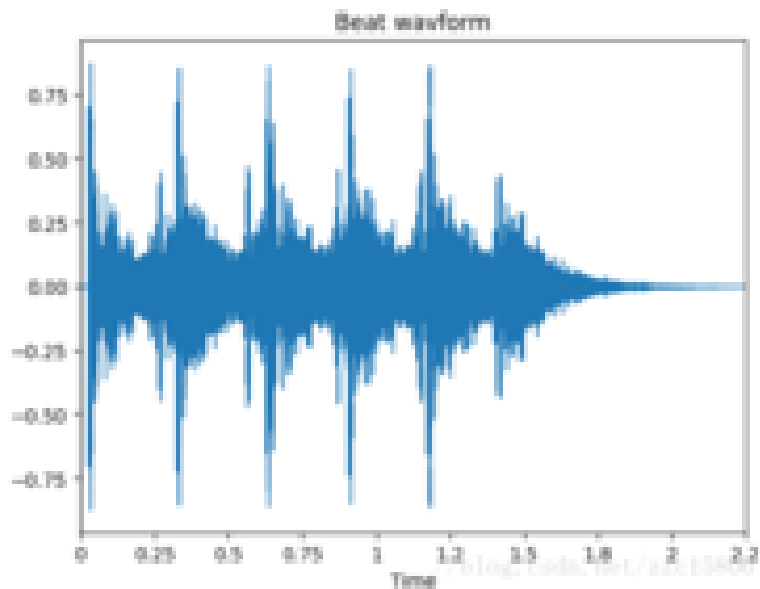
绘图显示

绘制声音波形

Librosa有显示声音波形函数waveplot():

```
1 >>> import librosa
2 >>> import librosa.display
3 >>> # Load a wav file
4 >>> y, sr = librosa.load('./beat.wav', sr=None)
5 >>> # plot a waveform
6 >>> plt.figure()
7 >>> librosa.display.waveplot(y, sr)
8 >>> plt.title('Beat waveform')
9 >>> plt.show()
```

输出图形为:

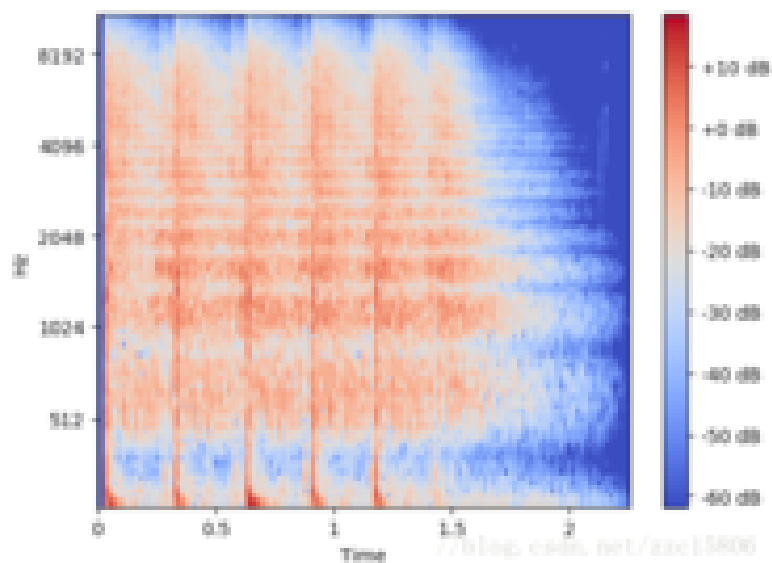


绘制频谱图

Librosa有显示频谱图波形函数specshow():

```
1 >>> import librosa
2 >>> import librosa.display
3 >>> # Load a wav file
4 >>> y, sr = librosa.load('./beat.wav', sr=None)
5 >>> # extract mel spectrogram feature
6 >>> melspec = librosa.feature.melspectrogram(y, sr, n_fft=1024, hop_length=512, n_mels=128)
7 >>> # convert to log scale
8 >>> logmelspec = librosa.power_to_db(melspec)
9 >>> # plot mel spectrogram
10 >>> plt.figure()
11 >>> librosa.display.specshow(logmelspec, sr=sr, x_axis='time', y_axis='mel')
12 >>> plt.title('Beat waveform')
13 >>> plt.show()
```

输出结果为:

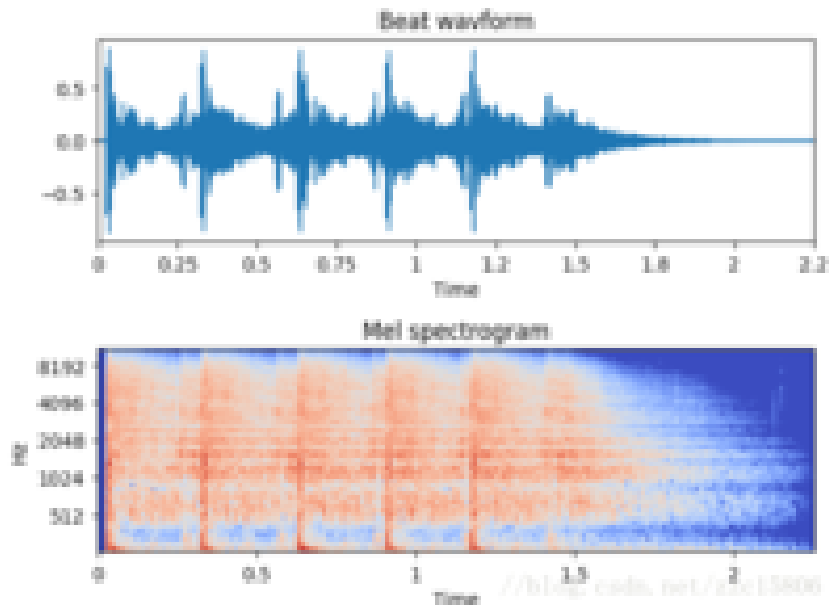


将声音波形和频谱图绘制在一张图表中:

```
1 >>> import librosa
2 >>> import librosa.display
3 >>> # Load a wav file
4 >>> y, sr = librosa.load('./beat.wav', sr=None)
5 >>> # extract mel spectrogram feature
```

```
6 | >>> melspec = librosa.feature.melspectrogram(y, sr, n_fft=1024, hop_length=512, n_mels=128)
7 | >>> # convert to log scale
8 | >>> logmelspec = librosa.power_to_db(melspec)
9 | >>> plt.figure()
10 | >>> # plot a waveform
11 | >>> plt.subplot(2, 1, 1)
12 | >>> librosa.display.waveplot(y, sr)
13 | >>> plt.title('Beat waveform')
14 | >>> # plot mel spectrogram
15 | >>> plt.subplot(2, 1, 2)
16 | >>> librosa.display.specshow(logmelspec, sr=sr, x_axis='time', y_axis='mel')
17 | >>> plt.title('Mel spectrogram')
18 | >>> plt.tight_layout() #保证图不重叠
19 | >>> plt.show()
```

输出结果为：



到这里，librosa的安装和简单使用就介绍完了。事实上，librosa远不止这些功能，关于librosa更多的使用方法还请大家参考librosa官网 <http://librosa.github.io/librosa/index.html>

参考：<http://librosa.github.io/librosa/index.html>



想对作者说点什么

使用librosa&SVM实现语言情感识别

阅读数 661

任务：语言情感分类音频处理库：librosalibrosa安装Librosa官网提供了多种安装方法，详细如下：最简单的方法就... 博文 来自：[as472780551的博客](#)

Librosa音频处理（一）

阅读数 1250

Librosa是一个用于音乐和音频分析的python包，如果没学过《数字信号处理》需要先了解一下相关的基础知识，傅... 博文 来自：[seTaire的博客](#)

【python】Windows中编译安装libsamplerate和scikits.samplerate

阅读数 2519

librosa缘由librosa是一个音频和音乐处理的Python包，我用它来做音频的特征提取。但是在使用时，发现librosa.lo... 博文 来自：[Jason Ding的专栏](#)

Python函数之librosa.load()函数

阅读数 725

librosa.load()函数用途：读取文件，可以是wav、mp3等格式。官方介绍：<https://librosa.github.io/librosa/gener...> 博文 来自：[qq_29884019的博客](#)

音频特征提取工具librosa

阅读数 601

前言 本文主要记录librosa工具包的使用，librosa在音频、乐音信号的分析中经常用到，是python的一个工具包，这... 博文 来自：[qq_21210467的博客](#)

python音频特征值提取librosa机器学习下载

python音频特征值提取librosa机器学习先将一段pcm格式的WAV文件进行解码，结果以0~1的double型，左右声道分别...

论坛

安装librosa遇到的问题		问答	
librosa audioread.NoBackendError		阅读数 2913	
最近在做语音合成librosa合成出来的音频，读取时候报错audioread.NoBackendError代码debug进去发现withaud... 博文 来自: qq_37175369的博客			
librosa, melspectrogram初阶		阅读数 1314	
importlibrosaimportlibrosa.display,sr=librosa.load("E:\\ML\\UrbanSound8K\\code\\UrbanSound8K\\audio... 博文 来自: c2c2c2aa的博客			
librosa的安装		阅读数 1735	
在很多设计到语音识别合成等方面的项目里经常用到python的一个包librosa但是这个包直接用pip安装容易出现GCC... 博文 来自: weixin_40128276...			
librosa		阅读数 59	
ubuntu系统、python2.7,安装librosa,需要joblib==0.11.0版本,版本不匹配可能会报librosa.TypeError:expectedstri... 博文 来自: zz_hh_uu的博客			
 <div>一个处女座的程序猿 1219篇文章 关注 排名:47</div>	 <div>骑着毛驴去旅行 12篇文章 关注 排名:千里之外</div>	 <div>qq7835144@163.com 16篇文章 关注 排名:千里之外</div>	 <div>seTaire 46篇文章 关注 排名:千里之外</div>
语音基音pitch的提取		07-17	
MATLAB程序 corr是求一个序列自相关的。 算法具有一般性。 zerocros求一个序列的过零次数 可以作为VAD的一个简单算法。 maxx是一个p... 下载			
音频特征提取——常用音频特征		阅读数 3050	
作者：桂。链接：http://www.cnblogs.com/xingshansi/p/6815217.html 前言主要总结一下常用的音频特征，并给... 博文 来自: 一摩尔自由的博客			
MFCC python plot		阅读数 1103	
#!/usr/bin/envpythonimportosfrompython_speech_featuresimportmfccfrompython_speech_featuresimport... 博文 来自: binqiang2wang			
基于MFCC参数的元音识别		阅读数 503	
基于MFCC参数的元音比对一、需求分析利用MFCC参数，对元音进行比对。读取每个元音的WAV文件，然后进行分... 博文 来自: 赵至柔的博客			
librosa 安装后遇到的问题		阅读数 37	
出现错误：FileNotFoundError:Nosuchfileordirectory:'avconv'解决方法：安装ffmpegubuntu16.04conda虚拟环... 博文 来自: weixin_37590425...			
python2安装librosa出现TypeError: expected string or buffer		阅读数 619	
版本不对应。需要卸载joblib0.12安装0.11pipuninstalljoblibpipinstalljoblib==0.11 博文 来自: qq_33266320的博客			
【论文导读】 Learning to Localize Sound Source in Visual Scenes		阅读数 747	
论文题目：LearningtoLocalizeSoundSourceinVisualScenes作者：ArdaSenocak,Tae-HyunOh,JunsikKim,Ming... 博文 来自: z小白的博客			
STFT和声谱图，梅尔频谱（Mel Bank Features）与梅尔倒谱（MFCCs）		阅读数 1087	
最近小编在做ASC（AcousticSceneClassification）问题，不管是用传统的GMM模型，还是用机器学习中的SVM或... 博文 来自: lbaihao的专栏			
人工智能学习图谱，学习AI的程序员需了解！			
如何能够短时间内抓住技术重点，打造属于自己的“offer收割机”？			
Ubuntu16.04下安装tensorflow（Anaconda3+pycharm+tensorflow+CPU）		阅读数 1万+	
GPU版本安装教程：https://blog.csdn.net/zzc15806/article/details/806527491.下载并安装Anaconda1.1下载从... 博文 来自: z小白的博客			
梅尔倒谱系数实现-MFCC		阅读数 790	
"""@author:zoutai@file:mymfcc.py@time:2018/03/26@description:"""fr... 博文 来自: SoundSlow的博客			
端到端的TTS深度学习模型tacotron(中文语音合成)		阅读数 1万+	
TACONTRON:AFullyEnd-to-EndText-To-SpeechSynthesisModel通常的TTS模型包含许多模块，例如文本分析， ... 博文 来自: yunnangf的博客			
python音频特征值提取librosa机器学习		06-07	
python音频特征值提取librosa机器学习先将一段pcm格式的WAV文件进行解码，结果以0~1的double型，左右声道分别存放。然后将16ms的... 下载			