语音信号处理之被掐死学习笔记

宋世杰 2023.11.14-2023.11.17

目录

一. 学会儿数学

语音信号处理的深度学习入门 - 知乎 (zhihu.com)

傅里叶分析之掐死教程（完整版）更新于 2014.06.06 - 知乎 (zhihu.com)

马同学 (matongxue.com)

形象易懂的傅里叶变换、短时傅里叶变换和小波变换 (qq.com)

快速理解 FFT 算法（完整无废话） - 知乎 (zhihu.com)

1.1 傅里叶级数

1.1.1 直接看理论

假设，$f(x)$为周期为T的函数，并且满足傅立叶级数的收敛条件，那么可以写作傅立叶级数：

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty}\left(a_n \cos(\frac{2\pi nx}{T}) + b_n \sin(\frac{2\pi nx}{T})\right)$$

其中：

$$a_n = \frac{2}{T}\int_{x_0}^{x_0+T} f(x)\cdot\cos(\frac{2\pi nx}{T})\,dx$$
$$b_n = \frac{2}{T}\int_{x_0}^{x_0+T} f(x)\cdot\sin(\frac{2\pi nx}{T})\,dx$$

----------复数域----------

根据欧拉公式：

$$e^{i\theta} = \cos\theta + i\sin\theta$$

我们可以推出：

$$\sin\theta = \frac{e^{i\theta} - e^{-i\theta}}{2i}$$
$$\cos\theta = \frac{e^{i\theta} + e^{-i\theta}}{2}$$

根据上式，我们可以写出傅立叶级数的另外一种形式：

$$f(x) = \sum_{n=-\infty}^{\infty} c_n \cdot e^{i\frac{2\pi nx}{T}}$$

其中：

$$c_n = \frac{1}{T}\int_{x_0}^{x_0+T} f(x)\cdot e^{-i\frac{2\pi nx}{T}}\,dx$$

1.1.2 深挖下逻辑

①任意周期函数可以分解为奇偶函数的和，而 sin 和 cos 正好一奇一偶。

但是任意函数可以分解和奇偶函数之和：

周期

$$f(x) = \frac{f(x)+f(-x)}{2} + \frac{f(x)-f(-x)}{2} = f_{\text{even}} + f_{\text{odd}}$$

所以同时需要$sin(x), cos(x)$。

②w=2π/T=1 时，周期是 2π，此时 nw 的周期也一定包含 2π（但不是最小周期）

更一般的，如果$f(x)$的周期为T，那么：

$$sin(\frac{2\pi n}{T}x) \quad cos(\frac{2\pi n}{T}x), n \in \mathbb{N}$$

这些函数的周期都为T。

将这些函数进行加减，就保证了得到的函数的周期也为T。

③f(x)可用小周期的奇偶函数相加，因此确定任意周期函数可以表示的形式。

综上，构造出来的三角函数之和大概类似下面的样子：

$$f(x) = C + \sum_{n=1}^{\infty}\left(a_n cos(\frac{2\pi n}{T}x) + b_n sin(\frac{2\pi n}{T}x)\right), C \in \mathbb{R}$$

这样就符合之前的分析：

- 有常数项
- 奇函数和偶函数可以组合出任意函数
- 周期为T
- 调整振幅，逼近原函数

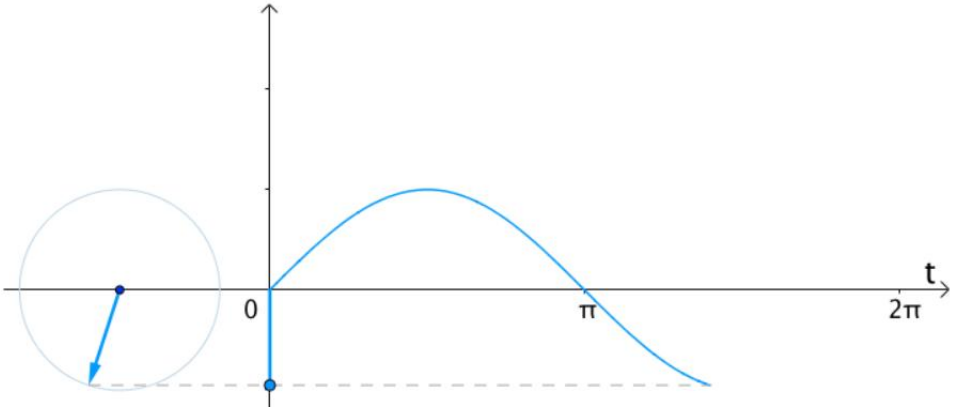
④换个角度：欧拉公式中 $e^{i\theta} \rightarrow e^{it}$ 可看做绕圆转，而它的虚部是 sin(t)，实部是 cos(t)。这两种角度，一个可以观察到旋转的频率，所以称为频域；一个可以看到流逝的时间，所以称为时域：

根据欧拉公式，有：

$$e^{it} = cos(t) + isin(t)$$

e^iwt，这里取了w=1

所以，在时间t轴上，把e^{it}向量的虚部（也就是纵坐标）记录下来，得到的就是$sin(t)$：



⑤欧拉公式形式和三角函数所表示的竟是两个线性空间，还是都希尔伯特空间！

希尔伯特空间：可以理解为无限维实向量空间。基是谁?看下图

我们令：

$$G(t) = e^{it} + e^{i2t}$$

这里用大写的G来表示复数函数。

刚才看到了，e^{it}和e^{i2t}都是向量，所以上式可以写作：

$$\vec{G(t)} = \vec{e^{it}} + \vec{e^{i2t}}$$

这里就是理解的重点了，从线性代数的角度：

- e^{it}和e^{i2t}是基（可以参考无限维的希尔伯特空间）
- $G(t)$是基e^{it}, e^{i2t}的线性组合

$g(t)$是$G(t)$的虚部，所以取虚部，很容易得到：

$$\vec{g(t)} = \vec{sin(t)} + \vec{sin(2t)}$$

即$g(t)$是基$sin(t), sin(2t)$的线性组合。

那么$sin(t), sin(2t)$的系数，实际上是$g(t)$在基$sin(t), sin(2t)$下的坐标了。

⑥希尔伯特空间下，可以通过点乘来求任意基的系数了，这样就实现了

函数向量的点积，可以由离散情况的点积（求和）迁移到连续的情况来。

在这里抛出一个结论（可以参考无限维的希尔伯特空间），函数向量的点积是这么定义的：

$$\vec{f(x)} \cdot \vec{g(x)} = \int_0^T f(x)g(x)dx$$

$$f(x) = C \cdot 1 + \sum_{n=1}^{\infty} \left(a_n cos(\frac{2\pi n}{T}x) + b_n sin(\frac{2\pi n}{T}x) \right), C \in \mathbb{R}$$

也就是说向量$f(x)$的基为：

两个不同的基函数点乘为0　　$\{1, cos(\frac{2\pi n}{T}x), sin(\frac{2\pi n}{T}x)\}$

是的，1也是基。

那么可以得到：　　分子分母都点乘其中某个基，其他基都成0了，所以就是求这个基下的系数

$$a_n = \frac{\int_0^T f(x)cos(\frac{2\pi n}{T}x)dx}{\int_0^T cos^2(\frac{2\pi n}{T}x)dx} = \frac{2}{T}\int_0^T f(x)cos(\frac{2\pi n}{T}x)dx$$

$$b_n = \frac{\int_0^T f(x)sin(\frac{2\pi n}{T}x)dx}{\int_0^T sin^2(\frac{2\pi n}{T}x)dx} = \frac{2}{T}\int_0^T f(x)sin(\frac{2\pi n}{T}x)dx$$

1.2 傅里叶变换

①由上节已经知道傅里叶级数下的空间和基，每个基其实对应频域上的一个频率。

$$f(x) \approx 1 + \frac{4}{\pi}sin(x) + 0sin(2x) + \frac{4}{3\pi}sin(3x) + 0sin(4x) + \frac{4}{5\pi}sin(5x)$$

此时基为：

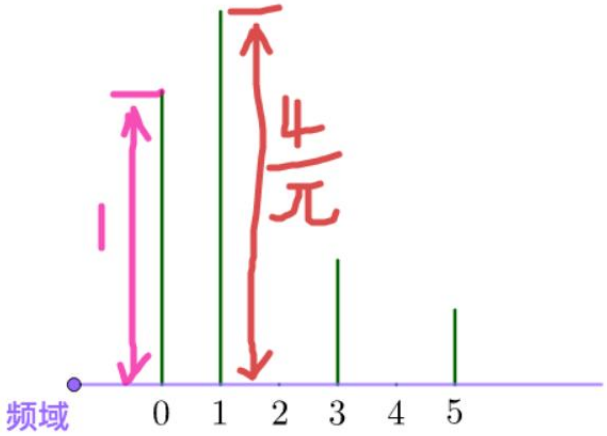
f(x)=向量*基1+向量*基2+...

$$\{1, sin(x), sin(2x), sin(3x), sin(4x), sin(5x)\}$$

对应的向量为：

$$(1, \frac{4}{\pi}, 0, \frac{4}{3\pi}, 0, \frac{4}{5\pi})$$

六维的向量没有办法画图啊，没关系，数学家发明了一个频域图来表示这个向量：

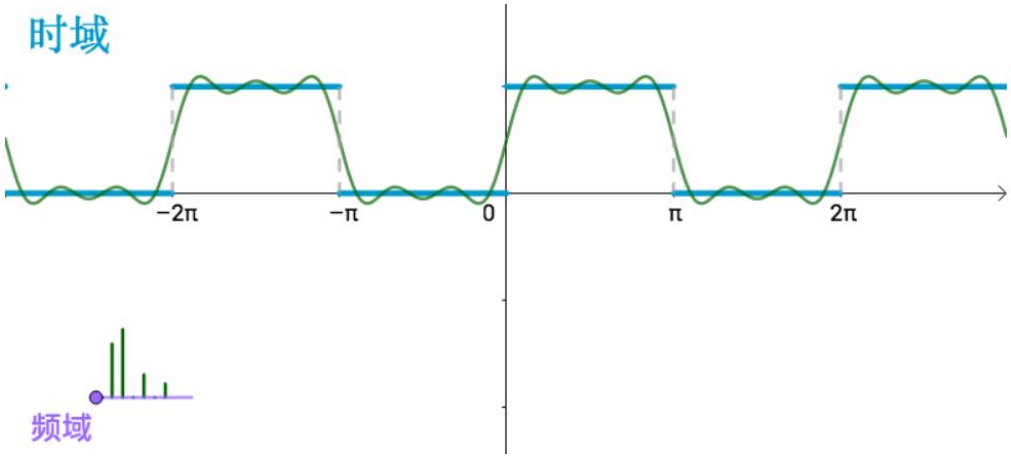


012345 分别代表了不同**频率**的正弦波函数，也就是之前的**基**：

$$0Hz \iff sin(0x) \quad 3Hz \iff sin(3x) \cdots$$

而高度则代表在这个频率上的**振幅**，也就是这个基上的坐标分量（理解为**基的坐标/向量长度** 或 傅里叶级数中的系数）。

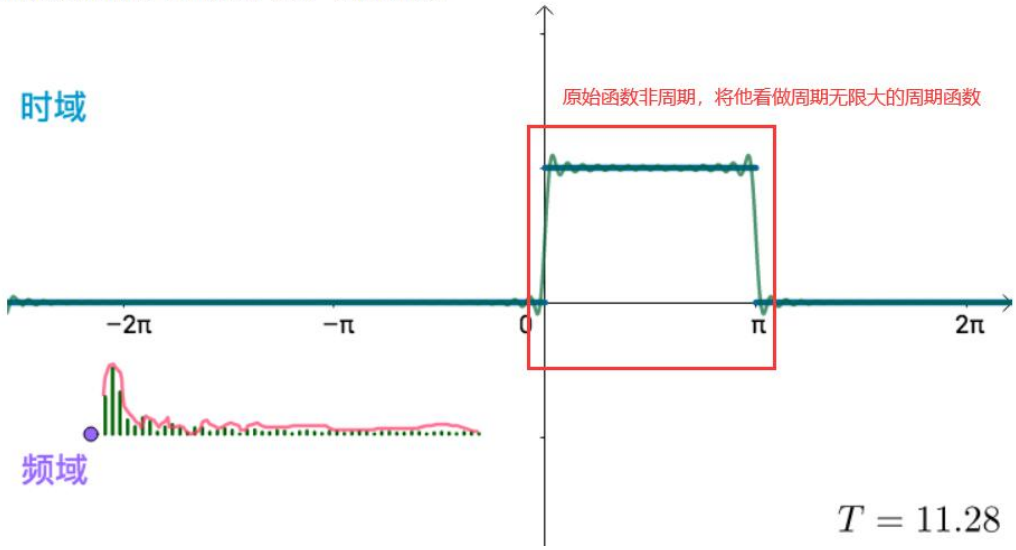
时域、频域其实反映的都是同一个曲线，只是一个是用函数的观点，一个是用向量的观点。



②周期函数—>非周期函数，无法傅里叶变换，那就将它视为无穷周期（T—>无穷）的函数，则频率 2π/(2πn/T) 为无限贴近且近 0，可以近似为一条连续曲线。

$$T = 2\pi \to T = \infty$$

这些频率就会变得稠密，直至连续，变为一条频域曲线：



③傅里叶变换

之前说了，傅立叶级数是：

$$f(x) = a_0 + \sum_{n=1}^{\infty}\left(a_n cos(\frac{2\pi n}{T}x) + b_n sin(\frac{2\pi n}{T}x)\right), a_0 \in \mathbb{R}$$

这里有正弦波，也有余弦波，画频域图也不方便，通过欧拉公式，可以修改为复数形式（请参考“代数细节”一文）：

其中：

$$f(x) = \sum_{n=-\infty}^{\infty} c_n \cdot e^{i\frac{2\pi nx}{T}}$$

$$c_n = \frac{1}{T}\int_{x_0}^{x_0+T} f(x)\cdot e^{-i\frac{2\pi nx}{T}}\,dx$$

复数形式也是向量，可以如下解读：

$$f(x) = \sum_{n=-\infty}^{\infty} \underbrace{c_n}_{\text{对应基的坐标}} \cdot \underbrace{e^{i\frac{2\pi nx}{T}}}_{\text{正交基}}$$

不过c_n是复数，不好画频域图，所以之前讲解全部采取的是三角级数。

周期推向无穷的时候可以得到：

$$\left.\begin{aligned} f(x) &= \sum_{n=-\infty}^{\infty} c_n \cdot e^{i\frac{2\pi nx}{T}} \\ T &= \infty \end{aligned}\right\} \Longrightarrow f(x) = \int_{-\infty}^{\infty} F(\omega)\, e^{i\omega x}\, d\omega$$

上面简化了一下，用ω代表频率。

$F(\omega)$大致是这么得到的：

$$\left.\begin{aligned} c_n &= \frac{1}{T}\int_{x_0}^{x_0+T} f(x)\cdot e^{-i\frac{2\pi nx}{T}}\, dx \\ T &= \infty \end{aligned}\right\} \Longrightarrow F(\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} f(x)\, e^{-i\omega x}\, dx$$

$F(\omega)$就是傅立叶变换，得到的就是频域曲线。

下面两者称为傅立叶变换对，可以相互转换：

$$f(x) \iff F(\omega)$$

正如之前说的，这是看待同一个数学对象的两种形式，一个是函数，一个是向量。

详解傅里叶变换的推导：

$c_n = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)e^{-jn\omega_0 t}dt$

$f(t) = \sum_{n=-\infty}^{n=+\infty} c_n e^{jn\omega_0 t}$

那么对于非周期函数，我们把它的周期看作无穷大。基频 $\omega_0 = \frac{2\pi}{T}$ 则趋近于无穷小，又因为基频相当于周期函数的傅里叶级数中两个相邻频率的差值 $(n+1)\omega_0 - n\omega_0$ ，我们可以把它记作 $\Delta\omega$ 或者微分 $d\omega$ 。 $n\omega_0$ 则相当于连续变量 ω 。这样就得到了针对非周期函数的频谱函数如下

代入原函数中的cn表达式中

$c_n = \frac{\Delta\omega}{2\pi}\int_{-\infty}^{+\infty} f(t)e^{-j\omega t}dt$

我们会发现这里的 c_n 是趋于0的。

将它代入$f(t) = \sum_{n=-\infty}^{n=+\infty} c_n e^{jn\omega_0 t}$

$f(t) = \sum_{-\infty}^{+\infty}(\frac{\Delta\omega}{2\pi}\int_{-\infty}^{+\infty} f(t)e^{-j\omega t}dt)e^{j\omega t} = \int_{-\infty}^{+\infty}\frac{1}{2\pi}(\int_{-\infty}^{+\infty} f(t)e^{-j\omega t}dt)e^{j\omega t}d\omega$

基

则非周期函数的傅里叶变换定义为

$F(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-j\omega t}dt$

我们可以发现 $c_n = \frac{d\omega}{2\pi}F(\omega)$ ，即我们选取了信号幅值在频域中的分布密度 $F(\omega)$ 来表示傅里叶变换，而不是相应频率的信号幅值大小 c_n 。因为如果选择后者，那傅里叶变换的函数值就都是无穷小了，这显然对我们没有任何帮助。

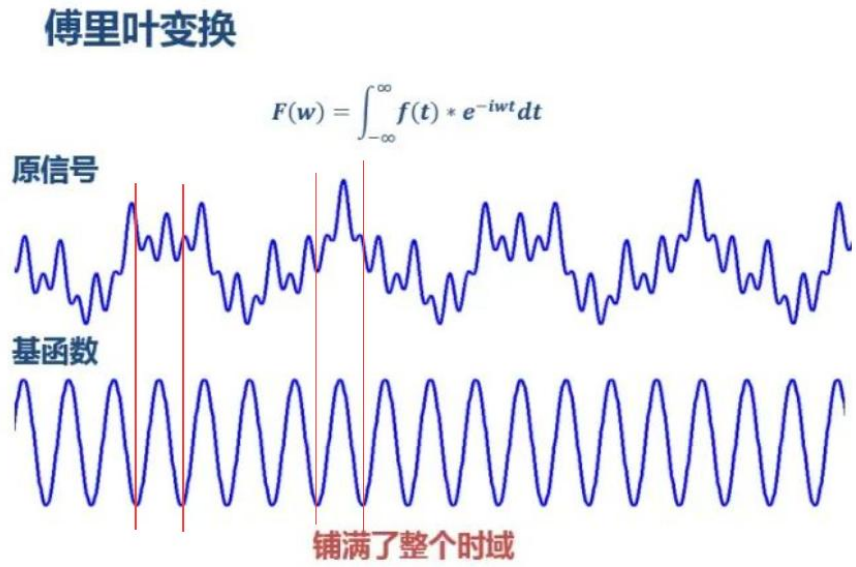
（信号幅值：C_n，幅值 C_n 的分布密度($\frac{P(x<\xi<x+\Delta x)}{\Delta x}$)就是傅里叶变换）

所以可以做总结，统计角度的傅里叶变换：

我们知道，**周期函数**的傅里叶级数实质上是将函数 $f(t)$ 分解为无数个不同频率、不同幅值的正、余弦信号，而这些信号的频率都是基频 ω_0 的整数倍（即 $n\omega_0$ ）。换言之，我们是在用无数个这样不同频率，不同幅值的正、余弦信号来逼近周期函数 $f(t)$ 。分解的过程中，对于每一个 $n\omega_0$ ，我们都得到了对应的幅值，这是不是就组成了一个函数关系（自变量为 $n\omega_0$ ，因变量为幅值，即相应频率信号的强度）？我们称之为**频谱函数**。

而对于**非周期函数**，傅里叶变换则是求**频谱密度函数**，该函数的自变量是 ω ，因变量是信号幅值在频域中的分布密度，即单位频率信号的强度。（**如果你学过概率论，可以将频谱函数和频谱密度函数类比为离散概率分布和概率密度函数**）

粗浅意义上的傅里叶变换：**计算信号和三角函数的相关性。**



1.3 离散傅里叶变换 DFT

（采样、时域、频域的离散化计算推导过程没看，直接记结论了）

对于傅里叶变换

$$F(f)=\int_{-\infty}^{+\infty} f(t)e^{-j2\pi ft}dt$$

我们做数学题时碰到的 $f(t)$ 大多数是在 t 上连续的，但由于计算机采集的信号在时域中是离散的，故实际应用中的 $f(t)$ 都是其经采样处理后得到的 $f_s(t)$ 。

同时，计算机也只可能计算出有限个频率上对应的幅值密度，即 f 也是离散的。

DFT就是 t 和 f 都为离散版的傅里叶变换。

连续信号 x(t)进行 N 次（N 为有限值）采样，得到 DFT 的表达式

上面讲到DFT的计算表达式为 $X[k]=\frac{1}{N}\sum_{n=0}^{N-1}x[n]e^{-j\frac{2\pi}{N}kn}$ ，复杂度为 $O(N^2)$ 。这里面的 n 相当于时域的 t ， k 相当于频率 $n\omega_0$ 中的 n ， $X[k]$ 则相当于我们之前说的频谱函数，表达的是频率为 $k\omega_0$ 时信号幅值的大小。

1.4 快速傅里叶变换 FFT

DFT的计算表达式为 $X[k]=\frac{1}{N}\sum_{n=0}^{N-1}x[n]e^{-j\frac{2\pi}{N}kn}$，复杂度为 $O(N^2)$

将 X[k]通过分成奇偶部分，探究其规律性，简化运算复杂度。

如何减小它的复杂度？能否利用 $e^{-j\frac{2\pi}{N}kn}$ 的周期性？通过观察可以发现，当n为偶数时（我们用2n表示）

$$e^{-j\frac{2\pi}{N}2nk}=e^{-j\frac{2\pi n}{N/2}k}=e^{-j\frac{2\pi n}{N/2}(k+N/2)}$$

我们令 $W[n,k]=e^{-j\frac{2\pi}{N}nk}$，由此可知当n为偶数时，$W[n,k]=W[n,k+N/2]$

将 $X[k]$ 分为n为偶数以及n为奇数两个部分。为了简便这里省去1/N。

$$X[k]=\sum_{n=0}^{N/2-1}x[2n]e^{-j\frac{2\pi}{N}2nk}+\sum_{n=0}^{N/2-1}x[2n+1]e^{-j\frac{2\pi}{N}(2n+1)k}$$

$$=\sum_{n=0}^{N/2-1}x[2n]e^{-j\frac{2\pi}{N}2nk}+e^{-j\frac{2\pi}{N}k}\sum_{n=0}^{N/2-1}x[2n+1]e^{-j\frac{2\pi}{N}2nk}$$

$$=\sum_{n=0}^{N/2-1}x[2n]e^{-j\frac{2\pi}{N/2}nk}+e^{-j\frac{2\pi}{N}k}\sum_{n=0}^{N/2-1}x[2n+1]e^{-j\frac{2\pi}{N/2}nk}$$

$$=\underbrace{\sum_{n=0}^{N/2-1}x[2n]W[2n,k]}_{\text{n为偶数部分}}+\underbrace{e^{-j\frac{2\pi}{N}k}\sum_{n=0}^{N/2-1}x[2n+1]W[2n,k]}_{\text{n为奇数部分}}$$

当n为奇数时，我们可以提取一个 $e^{-j\frac{2\pi}{N}k}$ 公因数，让剩下的 $W[n,k]$ 满足n为偶数的条件。

我们令偶数部分为 $E[k]$，奇数部分提取公因数后的结果为 $O[k]$，则

$X[k]=E[k]+W[1,k]O[k]$

$x[2n]$, $x[2n+1]$ 都是已知且固定的，且 $W[2n,k]$ 满足 $W[2n,k]=W[2n,k+N/2]$

那么，$E[k]=E[k+N/2]$，$O[k]=O[k+N/2]$。

$W[1,k]=-W[1,k+N/2]$ (推导如下)

$$e^{-j\frac{2\pi}{N}(k+N/2)}=cos(-\frac{2\pi}{N}(k+N/2))+jsin(-\frac{2\pi}{N}(k+N/2))=cos(-\frac{2\pi}{N}k-\pi)$$

$$+jsin(-\frac{2\pi}{N}k-\pi)=-e^{-j\frac{2\pi}{N}k}$$

我们可以得出结论

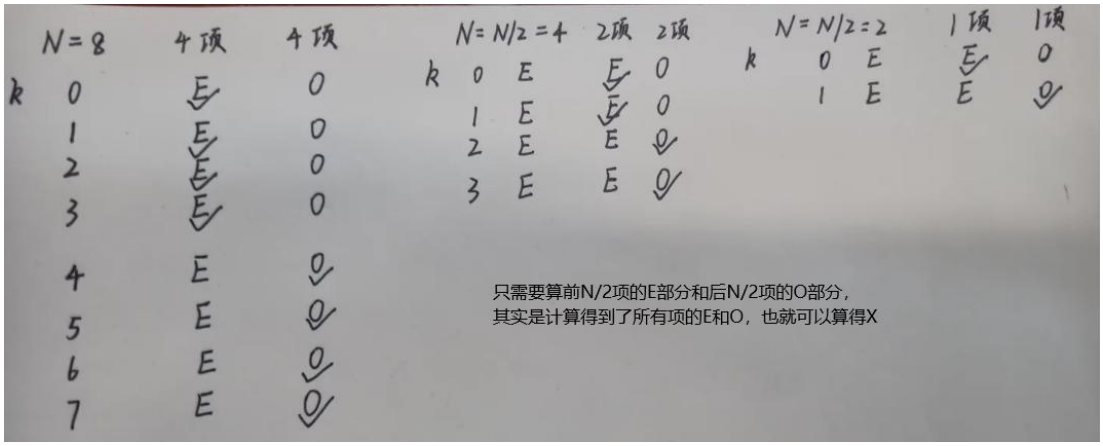
$X[k+N/2]=E[k]-W[1,k]O[k]$

E[0] + W[1,4] O[0]
=E[0] - W[1,0] O[0]

以N=8为例，$X[0]=E[0]+W[1,0]O[0]$, $X[4]=E[4]+W[1,4]O[4]$

也就是说，只要得到X[k]，我们必然可以得到X[k+N/2]。

算法过程：

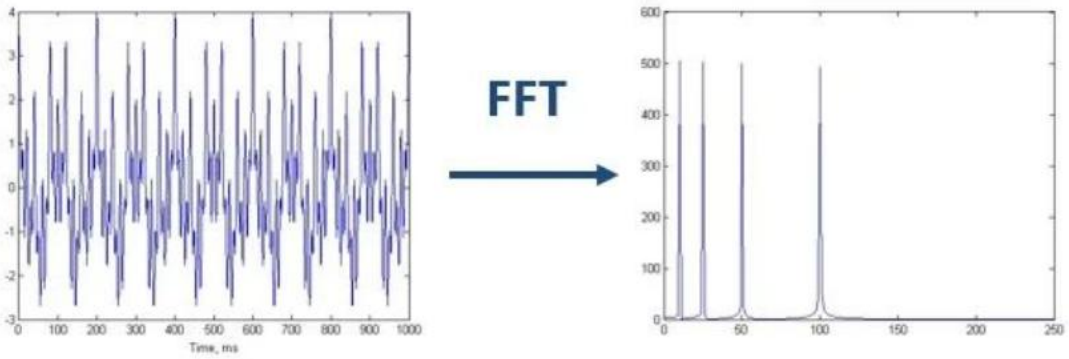


1.5 由 FFT 到短时傅里叶变换 STFT 和小波变换

形象易懂的傅里叶变换、短时傅里叶变换和小波变换（qq.com）

FFT 对非平稳信号处理具有局限性

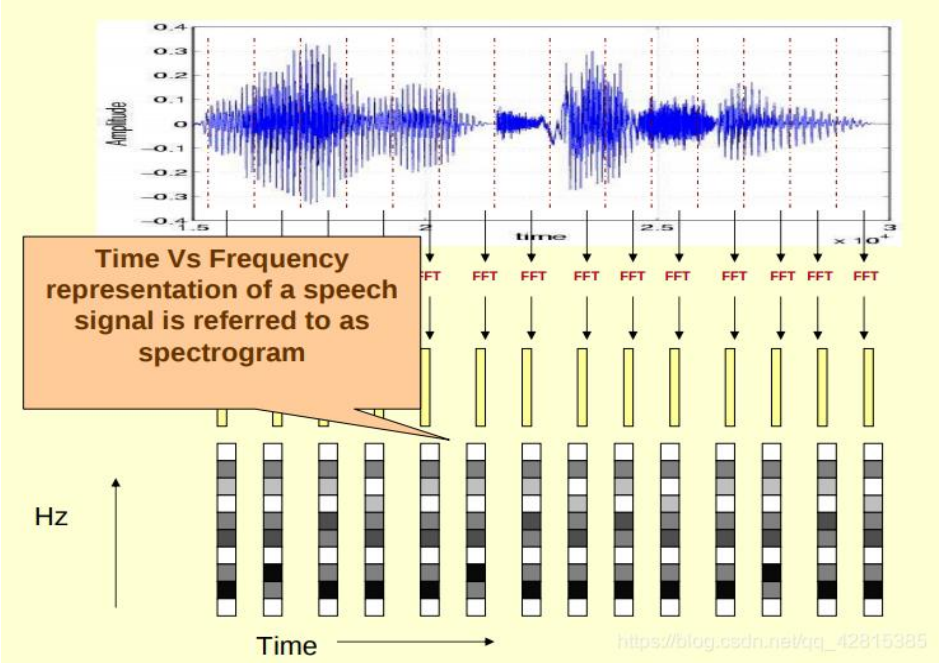
下面我们主要讲傅里叶变换的**不足。即我们知道傅里叶变化可以分析信号的频谱，那么为什么还要提出小波变换？**答案就是@方沁园所说的，**“对非平稳过程，傅里叶变换有局限性”**。看如下一个简单的信号：



$$x(t)=\cos(2\pi*10t)+\cos(2\pi*25t)+\cos(2\pi*50t)+\cos(2\pi*100t)$$

10, 25, 50, 100Hz

短时傅里叶变换 STFT：分帧加窗-FFT-合并



首先将将窗口移动到信号的开端位置，此时窗函数的中心位置在 $t = \tau_0$ 处，对信号加窗处理

$$y(t) = x(t) \cdot w(t - \tau_0)$$

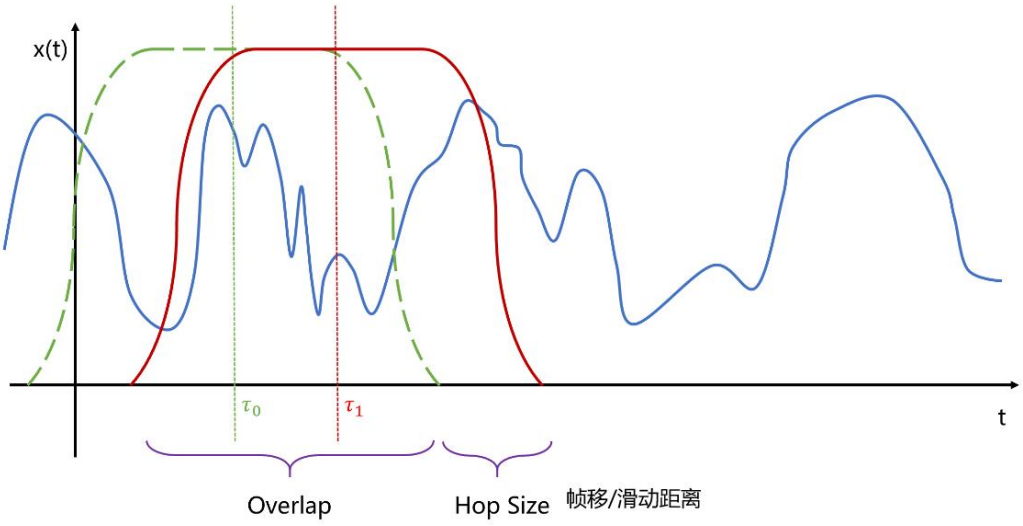
然后进行傅里叶变换

定义S（w,τ）= $X(\omega) = \mathcal{F}(y(t)) = \int_{\infty}^{+\infty} x(t) \cdot w(t - \tau_0) e^{-j\omega t} dt$

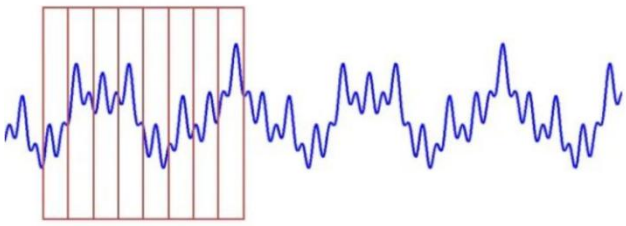
由此得到第一个分段序列的频谱分布 $X(\omega)$。在现实应用中，由于信号是离散的点序列，所以我们得到的是频谱序列 $X[N]$。

帧移

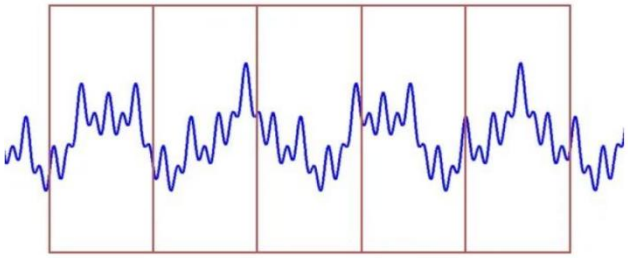
完成了对第一个分段的FFT操作后，移动窗函数到 τ_1。把窗体移动的距离称为 Hop Size。移动距离一般小于窗口的宽度，从而保证前后两个窗口之间存在一定重叠部分，我们管这个重叠叫 Overlap。



STFT 遇到的问题



框太窄 → 频率分辨率差

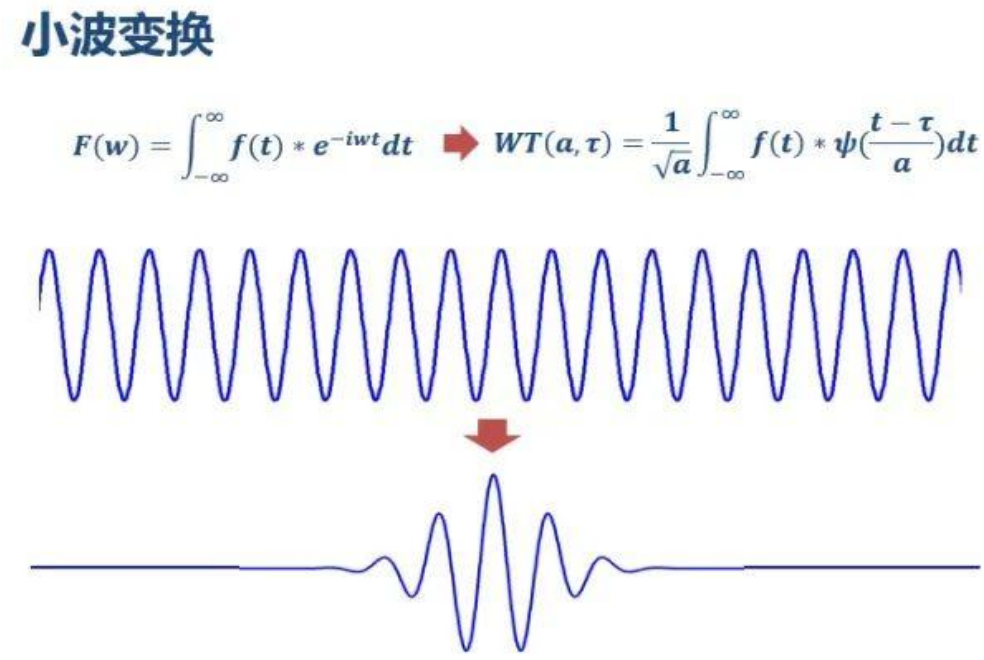


框太宽 → 时间分辨率差

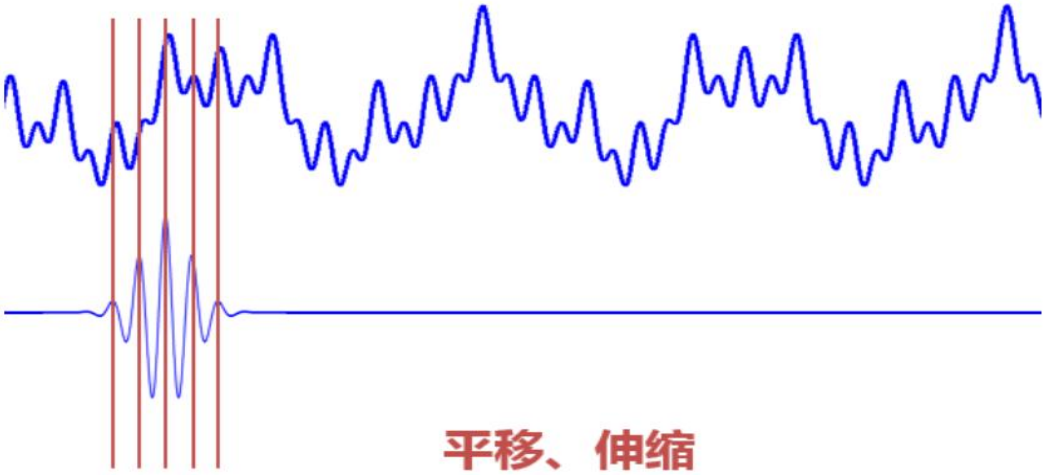
所以窄窗口**时间分辨率高、频率分辨率低**，宽窗口**时间分辨率低、频率分辨率高。**对于时变的非稳态信号，**高频适合小窗口，低频适合大窗口。**然而 STFT 的窗口是固定的，在一次 STFT 中宽度不会变化，所以 STFT 还是无法满足非稳态信号变化的频率的需求。

小波变换

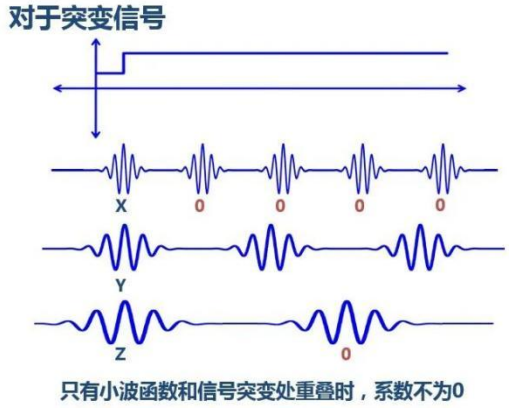
小波相对 STFT 的改变在，将无限长三角函数基换成了**有限长的会衰减的小波基**。



从公式可以看出，不同于傅里叶变换，变量只有频率ω，小波变换有两个变量：尺度a（scale）和平移量τ（translation）。**尺度**a控制小波函数的**伸缩**，**平移量** τ控制小波函数的**平移**。**尺度**就对应于**频率**（反比），**平移量** τ就对应于**时间**。



小波还有一些好处，比如，我们知道对于**突变信号**，傅里叶变换存在吉布斯效应，我们用无限长的三角函数怎么也拟合不好突变信号。



1.6 至此总结（自己总的，也可能有分析不对的地方）

傅里叶级数	时域上**连续的**周期函数 f(t) 分解为 **离散的**(nw_0, 幅值) 频谱函数
傅里叶变换 FT	时域上**连续的**非周期函数 f(t)（无限长延拓）变换为 频域上**连续的** F(w, 幅值的分布密度) 频谱密度函数
离散傅里叶变换 DFT	时域上**离散的**非周期函数 f(t)（无限长延拓）变换为 频域上**离散的** F(w, 幅值的分布密度) 频谱密度函数
快速傅里叶变换 FFT	同 DFT，但是减小了运算复杂度
短时傅里叶变换 STFT	针对非平稳信号，在 FFT 前增加了分帧加窗的操作
小波变换	针对非平稳信号，将 STFT 的三角函数基换为了小波基

二. 音频文件详解

【精选】音频处理——详解 PCM 数据格式_pcm 格式-CSDN 博客

音频重采样 - WELEN - 博客园 (cnblogs.com)

要将一段音频模拟信号转换为数字表示，包含如下三个步骤：

1、Sampling(采样)

2、Quantization(量化)

3、Coding(编码)

2.1 音频文件采样

采样处理，实际上就是让采样数据能够完全表示原始信号，且采样数据能够通过重构还原成原始信号的过程。

PAM：是一系列离散样本之的结果。

2.1.1 采样率

每秒钟的样本数也被称之为采样率，采样率的单位用 Hz 表示，例如 1Hz 表示每秒钟对原始信号采样一次，1KHz 表示每秒钟采样 1000 次。

根据场景的不同，采样率也有所不同，采样率越高，声音的还原程度越高，质量就越好，同时占用空间会变大。例如：通话时的采样率为 8KHz，常用的媒体采样率有 44KHz，对于一些蓝光影片采样率高达 1MHz。

采样率（1s 采几个点） * 语音时长 = 语音采样点数

采样点 // 帧移长度（每次移动几个采样点） + 1 = 帧数（一帧有多个点）

例如有一个语音，某个录音机器以 16000 个点/秒的速度（采样率）去测试这个声音， 测了 2s，我们则会得到 2 * 16000 = 32000 个采样点。也就是说，实际在图片里看到的曲线虽然是连续的曲线，但是实际上是由 离散的 有间隔的点组成的。

2.2.2 重采样

一些工作的需要，需要保存成 FLV 文件，而在保存的过程中，48000 的采样率并不符合用 FLV 的封装标准（最高 44100），所以有时候需要通过调用如 ffmpeg 来重采样 pcm，并保存文件。

重采样分为上采样和下采样，下采样时需要对信号进行抽取，上采样时需要对信号进行插值。

1、信号的抽取

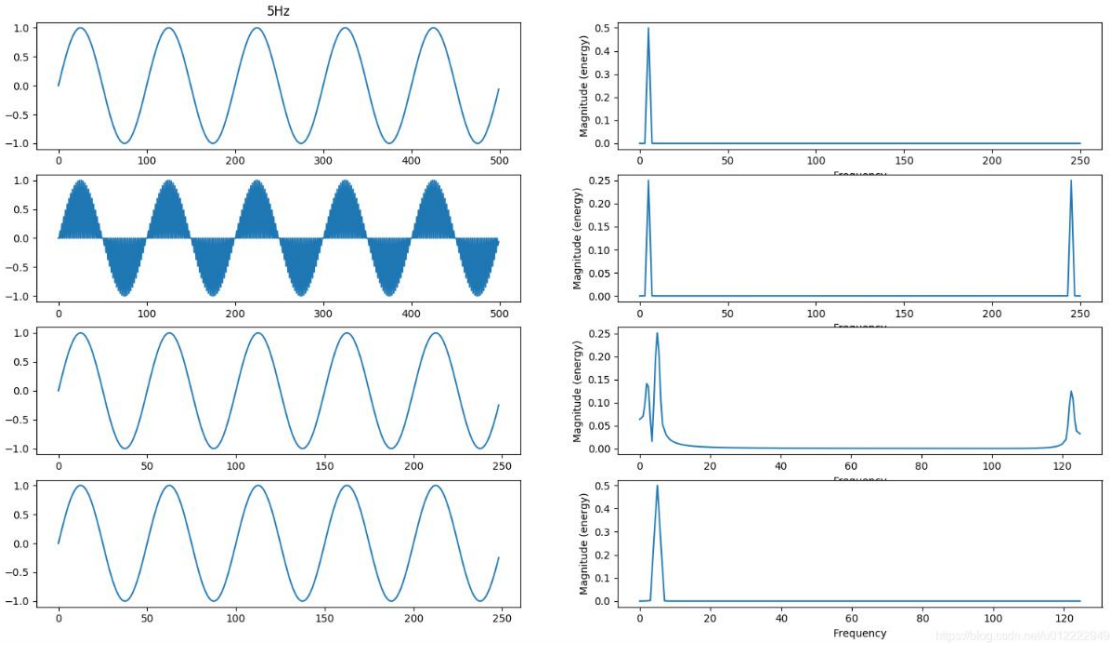
设$x(n)$为数字信号，欲使f_s减少M倍，最简单的方法是将$x(n)$中的每个点中抽取一个，依次组成一个新的序列$y(n)$，即

$$y(n) = x(Mn) \quad n \geq 0$$

此时，$y(n)$和$x(n)$的DTFT有如下关系（详见附A）：

$$Y(e^{jw}) = \frac{1}{M}\sum_{k=0}^{M-1} X(e^{j(w-2\pi k)/M})$$

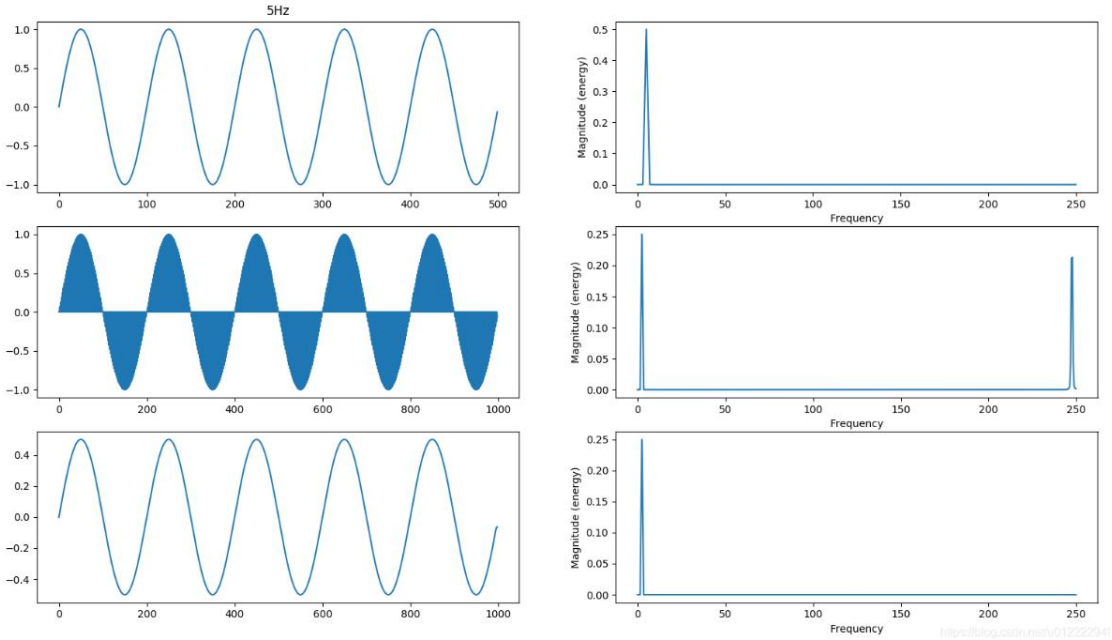
其含义是，将信号$x(n)$做M倍的抽取后，所得信号$y(n)$的频谱等于原信号$x(n)$的频谱先做M倍的扩展，再在w轴上做$\frac{2\pi}{M}$ $(k = 1, 2, ..., M-1)$的移位后再迭加。如下图所示。



2、信号的插值

将$x(n)$的采样率f_s增加L倍，即Lf_s，最简单的方法就是将$x(n)$每两个点之间补上$L-1$个零。设补零后的信号为$v(n)$，则

$$v(n) = \begin{cases} x(n/L) & n = 0, \pm L, \pm 2L, ... \\ 0 & others \end{cases}$$

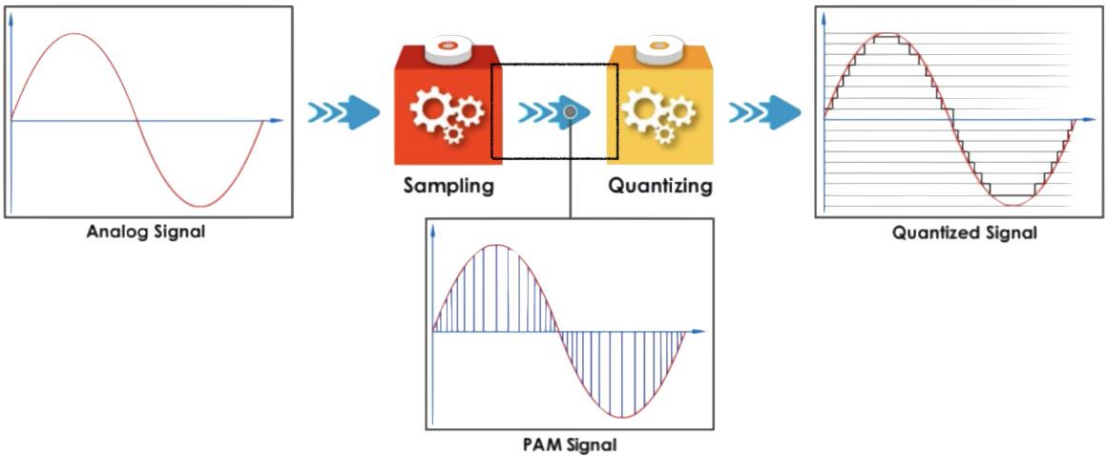


2.2 音频文件量化

原始信号采样后，需要通过量化来描述采样数据的大小。

量化处理过程，就是将时间连续的信号，处理成时间离散的信号，并用实数表示。这些实数将被转换为二进制数用于模拟信号的存储和传输。

在图例中，如果说采样是画垂直线段的话，那么量化就是画水平线，用于衡量每次采样的数字指标。



为了更好的描述量化过程，先来介绍一下 bit-depth（位深）：用来描述存储数字信号值的 bit 数。较常用的模拟信号位深有：

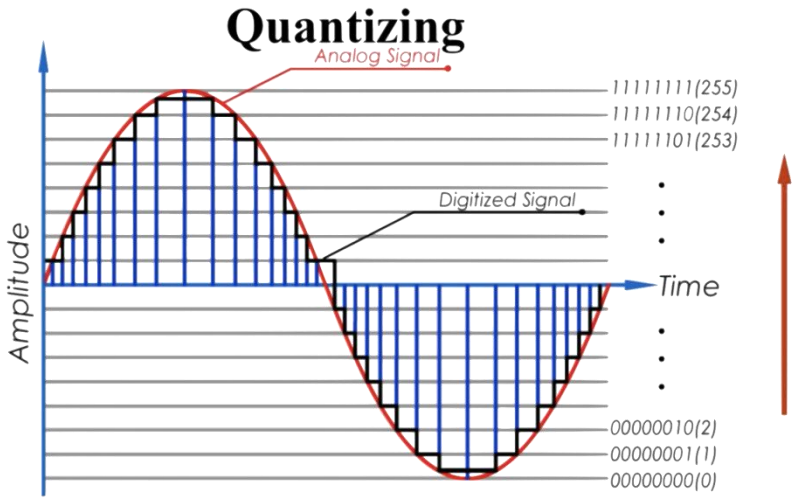
8-bit：2^8 = 256 levels，有 256 个等级可以用于衡量真实的模拟信号。

16-bit：2^16 = 65,536 levels，有 65,536 个等级可以用于衡量真实的模拟信号。

24-bit：2^24 = 16,666,216 levels，有更多个等级可以用于衡量真实的模拟信号。

显而易见，位深越大，对模拟信号的描述将越真实，对声音的描述更加准确。

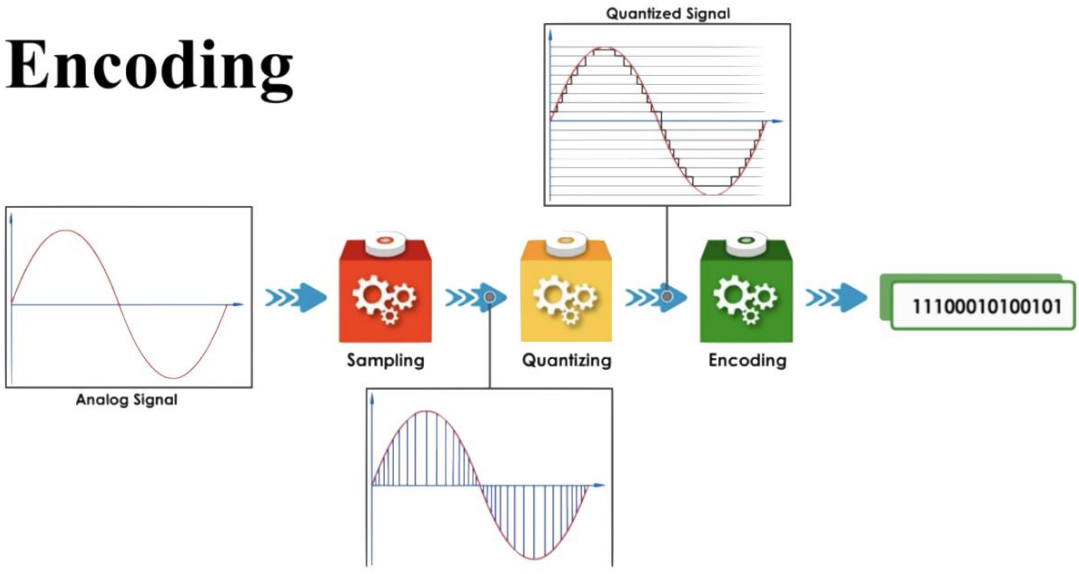
在当前例子中，如果用为 8-bit 位深来描述的话，就如下图所示：



量化的过程就是将一个平顶样本四舍五入到一个可用最近 level 描述的过程。如图中黑色加粗梯形折线。量化过程中，我们将尽量让每个采样和一个 level 匹配，因为每个 level 都是表示一个 bit 值。

图中，第 9 次采样的平顶样本对应的 level 用十进制表示为 255，也就是二进制的 1111 1111。

2.3 音频文件编码



在编码这一步，将时间线上的每个 sample 数据转化为对应的二进制数据。

在音视频中，PCM 是一种用数字表示采样模拟信号的方法。

采样数据经过编码后产生的二进制数据，就是 PCM 数据。PCM 数据可以直接存储在介质上，也可以在经过编解码处理后进行存储或传输。

PCM 常用指标：

• **采样率(Sample rate)：** 每秒钟采样多少次，以 Hz 为单位。

• **位深度(Bit-depth)：** 表示用多少个二进制位来描述采样数据，一般为 16bit。

• **字节序：** 表示音频 PCM 数据存储的字节序是大端存储（big-endian）还是小端存储（little-endian），为了数据处理效率的高效，通常为小端存储。

•**声道数（channel number）：** 当前 PCM 文件中包含的声道数，是单声道（mono）、双声道（stereo），此外还有 5.1 声道（常用于影院立体环绕声）等。

• **采样数据是否有符号（Sign）：** 要表达的就是字面上的意思，需要注意的是，使用有符号的采样数据不能用无符号的方式播放。

以 FFmpeg 中常见的 PCM 数据格式 s16le 为例：它描述的是有符号 16 位小端 PCM 数据。s 表示有符号，16 表示位深，le 表示小端存储。

大同小异，写一下常见音频文件读取示例（.wav 文件）：

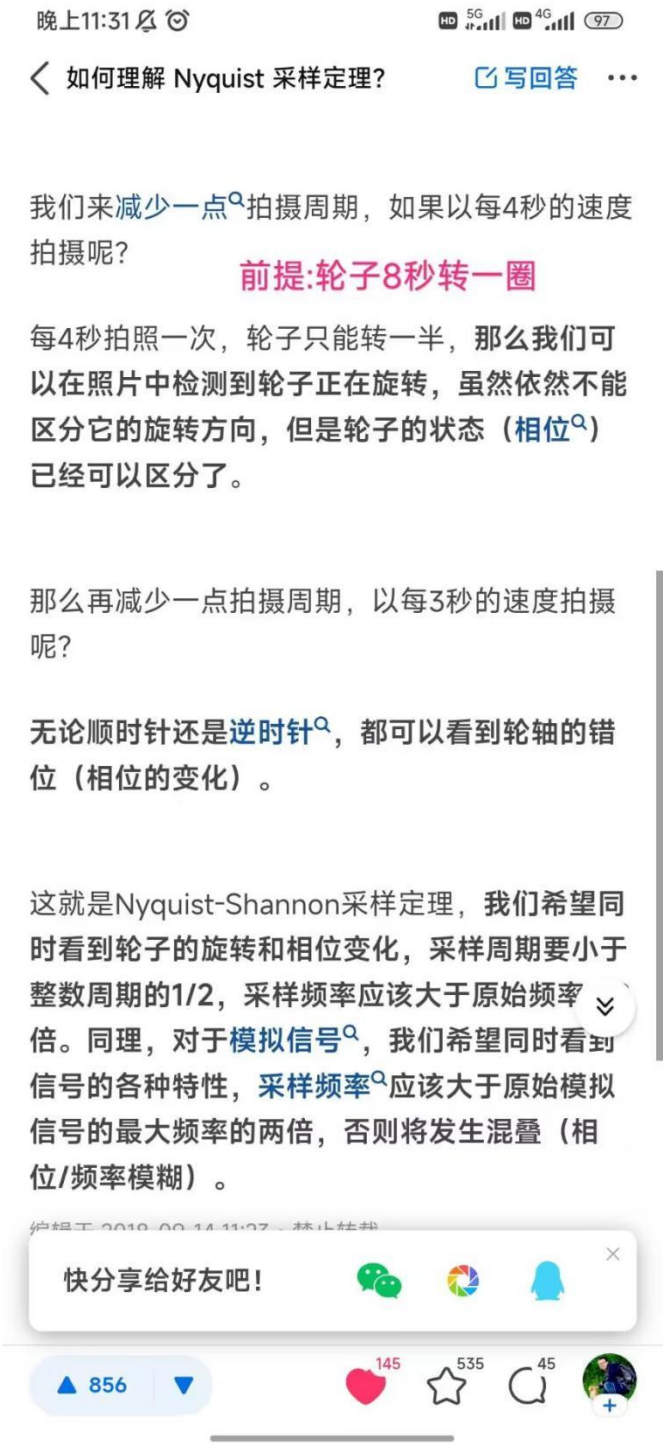
```
In[3]: wavfile="D:\桌面\心脏病识别\heartVentricle\data\heartbeat\\0_Normal\\96E0EB5077.wav"
In[4]: import os
  ...: import wave
  ...: import numpy as np
In[5]: wf = wave.open(wavfile, "rb")# 打开指定的WAV文件，使用 wave.open() 函数打开，打开模式为二进制读取（"rb"）
In[6]: params = wf.getparams()  #获取WAV文件的参数
In[7]: nchannels, sampwidth, framerate, nframes = params[:4] #通道数、采样宽度、帧率和采样点数，被存储在 params 变量中
In[8]: str_data = wf.readframes(nframes)
In[9]: wf.close()#从WAV文件中读取所有音频帧数据，存储在str_data变量中#关闭wav文件
In[10]: wave_data = np.frombuffer(str_data, dtype=np.short)# 将二进制数据转换为NumPy数组，数据类型为短整型（16位有符号整数）
In[11]: wave_data.shape = (-1, nchannels) #将其转换为二维数组  其中每行表示一个声道的数据？
In[12]: wave_data = wave_data.T # 将数组进行转置，使得转置后每行表示一个通道的数据
In[13]: print(wave_data)
[[  0   0   0 ... 132 114 108]]
```

```
▼ params = {_wave_params: 6} _wave_params(nchannels=1, sampwidth=2, framerate=8000, nframes=158720, comptype='NONE', compname='not compressed'
    compname = {str} 'not compressed'
    comptype = {str} 'NONE'
    framerate = {int} 8000
    nchannels = {int} 1
    nframes = {int} 158720
    sampwidth = {int} 2
    0 = {int} 1
    1 = {int} 2
    2 = {int} 8000
    3 = {int} 158720
    4 = {str} 'NONE'
    5 = {str} 'not compressed'
    __len__ = {int} 6
  ▶ Protected Attributes
  sampwidth = {int} 2
▶ str_data = {bytes: 317440} b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x0...
▶ wave_data = {ndarray: (1, 158720)} [[ 0  0  0 ... 132 114 108]] ...View as Array
```

2.4 采样定理

(7 封私信 / 33 条消息) 如何理解 Nyquist 采样定理？ - 知乎 (zhihu.com)

在进行模拟/数字信号的转换过程中，当采样频率 fs.max 大于信号中最高频率 fmax 的 2 倍时(fs.max>2fmax)，采样之后的数字信号完整地保留了原始信号中的信息，一般实际应用中保证采样频率为信号最高频率的 2.56～4 倍；

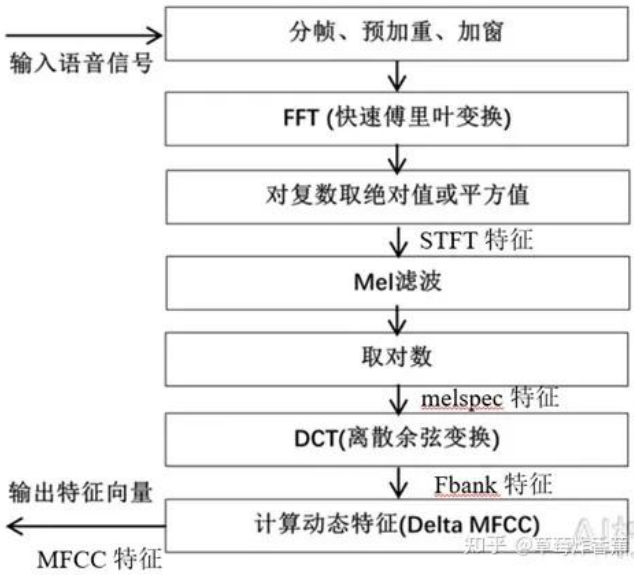


三. 语音信号处理（入门）

3.1 语音信号处理常规流程

[语音信号处理中怎么理解分帧？ - 知乎 (zhihu.com)](#)

[声谱图，梅尔谱图_梅尔声谱图-CSDN博客](#)（没会员没法看）



分帧

语音信号处理常常要达到的一个目标，就是弄清楚语音中**各个频率成分的分布**。做这件事情的数学工具是**傅里叶变换**。傅里叶变换要求输入信号是**平稳**的，当然不平稳的信号你想硬做也可以，但得到的结果就没有什么意义了。而语音在宏观上来看是不平稳的——你的嘴巴一动，信号的特征就变了。但是从微观上来看，在比较短的时间内，嘴巴动得是没有那么快的，语音信号就可以看成平稳的，就可以截取出来做傅里叶变换了。这就是为什么语音信号要分帧处理，截取出来的一小段信号就叫一「帧」。

如下图：这段语音的前三分之一和后三分之二明显不一样，所以整体来看语音信号不平稳。红框框出来的部分是一帧，在这一帧内部的信号可以看成平稳的。

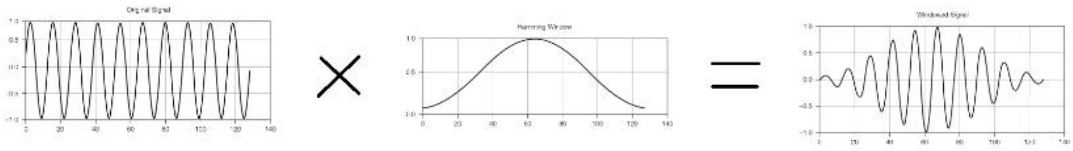
那么一帧有多长呢？帧长要满足两个条件：

- 从宏观上看，它必须足够短来保证帧内信号是平稳的。前面说过，口型的变化是导致信号不平稳的原因，所以在一帧的期间内口型不能有明显变化，即一帧的长度应当小于一个**音素**的长度。正常语速下，音素的持续时间大约是 50~200 毫秒，所以帧长一般取为小于 50 毫秒。
- 从微观上来看，它又必须包括足够多的振动周期，因为傅里叶变换是要分析频率的，只有重复足够多次才能分析频率。语音的基频，男声在 100 赫兹左右，女声在 200 赫兹左右，换算成周期就是 10 毫秒和 5 毫秒。既然一帧要包含多个周期，所以一般取至少 20 毫秒。

这样，我们就知道了帧长一般取为 20 ~ 50 毫秒，20、25、30、40、50 都是比较常用的数值，甚至还有人用 32（在程序猿眼里，这是一个比较「整」的数字）。

加窗（与分帧同步进行，分帧后加窗，然后帧移到下一窗口）

取出来的一帧信号，在做傅里叶变换之前，要先进行「**加窗**」的操作，即与一个「窗函数」相乘，如下图所示：

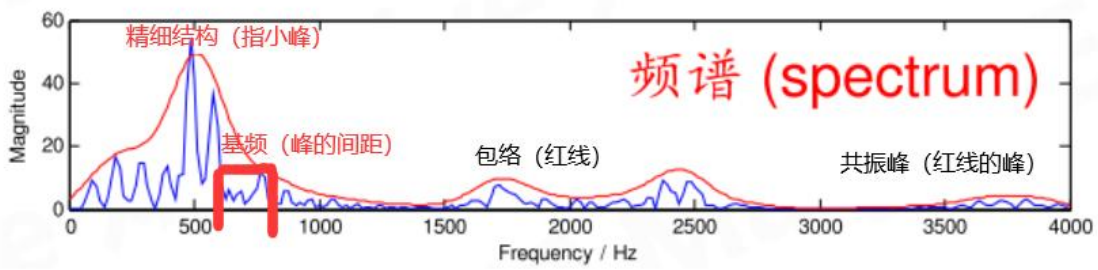


加窗的目的是让一帧信号的幅度在两端**渐变**到 0。渐变对傅里叶变换有好处，可以让频谱上的各个峰更细，不容易糊在一起（术语叫做**减轻频谱泄漏**），具体的数学就不讲了。

加窗的代价是一帧信号两端的部分被削弱了，没有像中央的部分那样得到重视。弥补的办法是，帧不要背靠背地截取，而是相互重叠一部分。相邻两帧的起始位置的时间差叫做**帧移**，常见的取法是取为**帧长的一半**，或者固定取为 **10 毫秒**。

傅里叶变换

对一帧信号做傅里叶变换，得到的结果叫**频谱**，它就是下图中的蓝线：



图中的横轴是频率，纵轴是幅度。频谱上就能看出这帧语音在 480 和 580 赫兹附近的能量比较强。语音的频谱，常常呈现出「精细结构」和「包络」两种模式。「精细结构」就是蓝线上的一个个小峰，它们在横轴上的间距就是**基频**，它体现了语音的**音高**——峰越稀疏，基频越高，音高也越高。「包络」则是连接这些小峰峰顶的平滑曲线（红线），它代表了口型，即发的是哪个音。包络上的峰叫**共振峰**。图中能看出四个，分别在 500、1700、2450、3800 赫兹附近。有经验的人，根据共振峰的位置，就能看出发的是什么音。

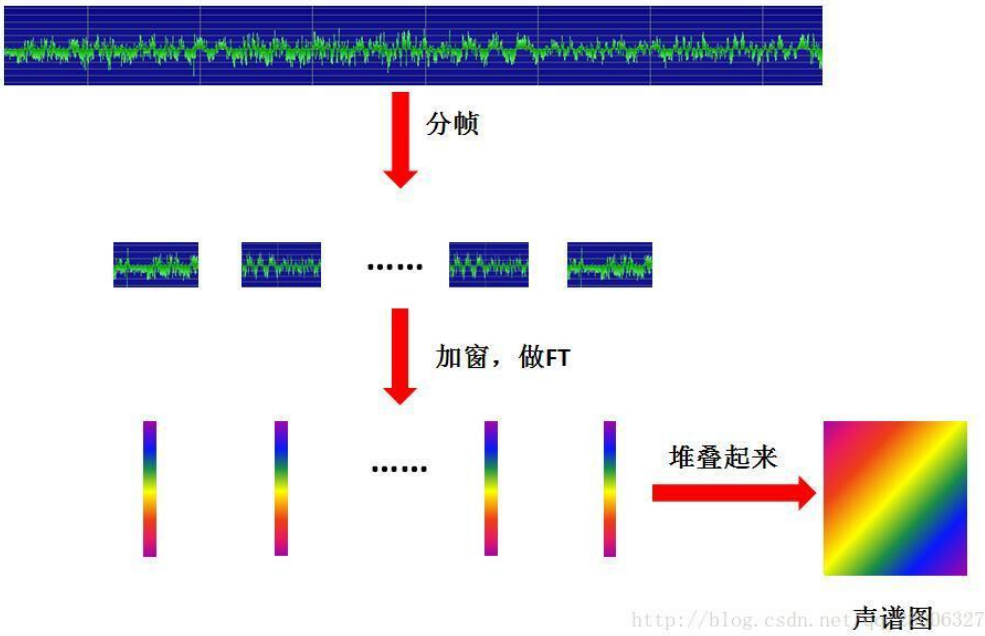
对每一帧信号都做这样的傅里叶变换，就可以知道音高和口型随时间的变化情况，也就能识别出一句话说的是什么了。

取特征

一个频谱转换为一个 39 维的特征向量序列。

短时傅里叶变换(STFT)

短时傅里叶变换(STFT)，就是对短时的信号做傅里叶变换。原理如下：对一段长语音信号，**分帧、加窗，再对每一帧做傅里叶变换**，之后把每一帧的结果沿另一维度堆叠，得到一张图（类似于二维信号），这张图就是声谱图。



STFT 算法，它对于一个表示为 1 行，T 列的信号（1，T），通常会设定一组线性增加的频率 ，如从 10hz，20hz，30hz…增长到 3Khz，接下来就假定信号由这些频率的三角函数信号叠加组成。这些三角函数，用时域的信号（波形）是很不好计算的，但是由另一个数域却能变得很好计算，那就是复数域。因此，FFT 计算是**先将 傅里叶级数变换到复数域，经过计算再变成时域**，此时，得到的结果是每个假定的三角函数信号的 一个复数表示，即为 a+bj。

我们可以得到两个矩阵，幅度（**300,frames**）（10hz,20hz...等距切割的的频率，帧数）和 相位（300,frames），前者即称之为**幅度谱（语谱图）**，后者称之为**相位谱**。通常来说，在 python 库的设定里 n_fft 即为对多少个信号点做傅里叶变换。

我们得到了 STFT、Mel 谱等等特征以后，就会将这些特征送入神经网络模型去学习其内在规律，从而实现语音识别、语音合成、音色转换、说话人识别、语音降噪、语音端点检测等等的任务！因此，学习信号处理是进一步学习深度学习的基础。

3.2 常见音频信号频谱特性

声波，有频率和振幅，频率高低决定音调，振幅大小决定响度，采样率（每秒所采样样本的总数目）是对频率采样，位深（每个样本中信息的比特数 16bit、24Bit）是对振幅采样。

（1）常见采样频率

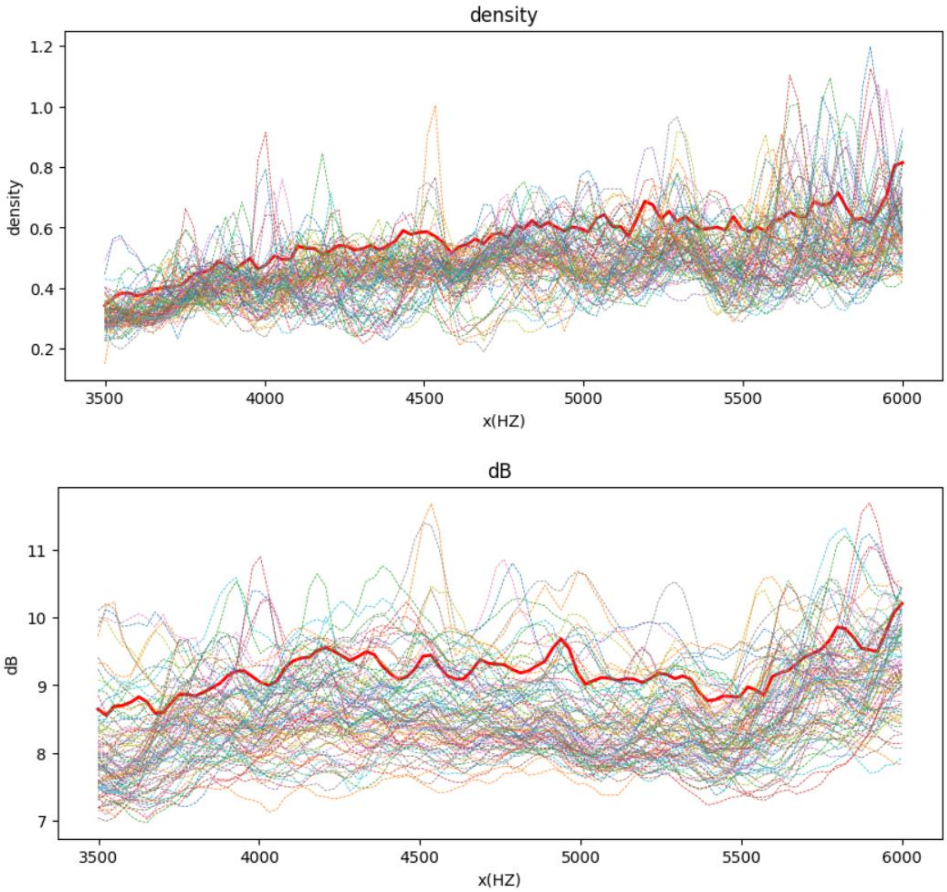
8000Hz： 电话所用采样率，这对于听清电话里的人声已经足够。

22.050KHz： 无线电广播所用采样率，广播音质。

鸭蛋检测：25000 HZ（实际）——>640000 HZ（算法默认）

心脏病检测：8000HZ

（2）频率-密度/能量曲线。（鸭蛋中裂纹蛋的包络线与所有好蛋）



（3）常见频率

以水管为例直观去感受频率

管子整体振动——频率低频段（长波）——传的远——穿透强——易于检测

分子间的撞击——频率高频段（短波）——损耗快——穿透弱——难以检测

优化：①增加听筒灵敏度 ②贴近待测物体去检测，避免过多损耗

人的说话声：300-3400Hz（考虑语音信号的能量集中部分，这个数据一般是通过考察元音的前三个共振峰而得到的一个范围）声强 10-50dB。

人耳可感受的声音频率范围在 16～20000Hz 之间，大于 20000Hz 的声音是超声，低于 16Hz 的声音是次声，，这两种声音人耳都感受不到。

4G 的频率和频段是：1880-1900MHz、2320-2370MHz、2575-2635MHz。

5G 的频率和频段频段：3300-3400MHz（原则上限室内使用）、3400-3600MHz 和 4800-5000MHz

同频共振原理：相同频率的物体会相互发生共鸣。当人体遇到与身体类似频率的振动噪音，身体也会同时发生振动，从而产生危害。所以军事上也会有很多次声波武器，其杀伤力惊人。

正常人体的共振频率应为 7.5Hz 左右，其中各部分又有自己的共振频率。如内脏为 4～6Hz，头部为 8～12Hz 等。生活中大部分低频噪音在 250Hz 左右。

四. 语音识别技术

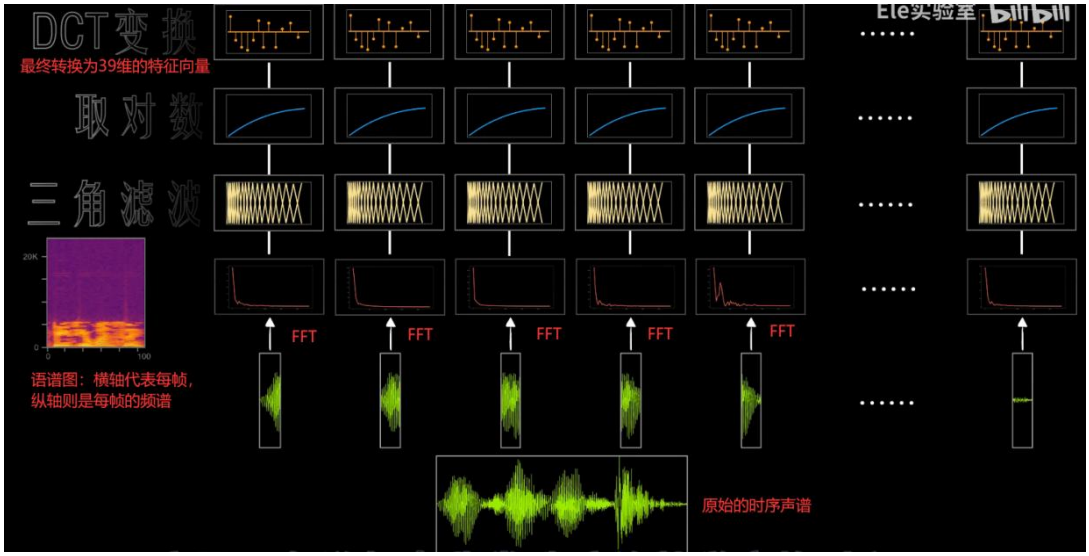
【语音识别技术】重度鉴赏_哔哩哔哩_bilibili

高斯混合模型（GMM） - 知乎 (zhihu.com)

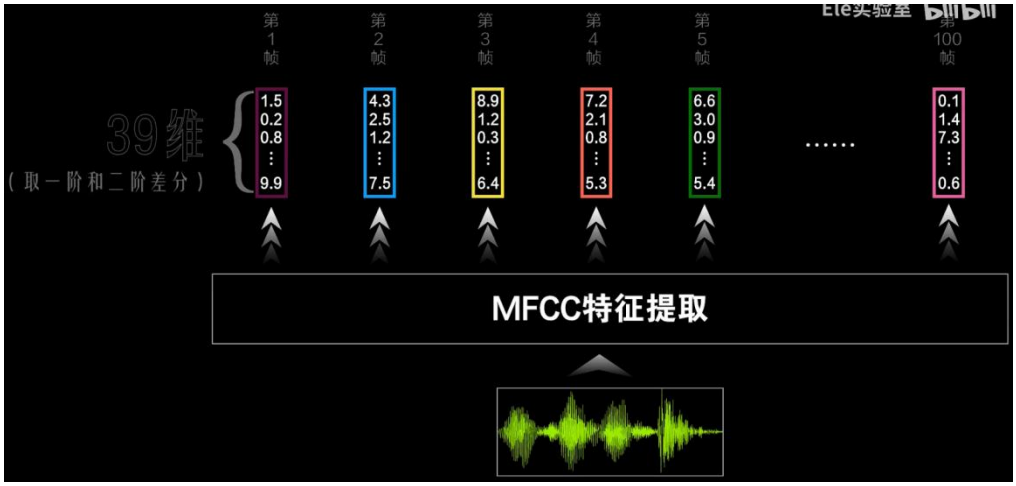
HMM 隐马尔可夫模型的例子、原理、计算和应用 - 知乎 (zhihu.com)

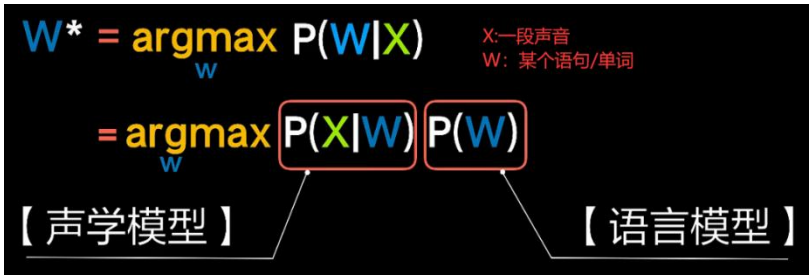
4.1 语音识别技术演变

（一）MFCC：从原始采集到的时序的声音，转换为一帧帧在频域上的特征向量。



以上仅是 MFCC 特征提取的过程，得到每帧对应的 39 维特征向量 lgdB





要对一段语音进行分类，即计算该声音是某个词的概率 P(W 某个词|X 声音)，则需要贝叶斯定律进行转换，同时利用以上提到的声学模型和语言模型。

（二）**声学模型：**每个词都对应一个模型，分辨一段声音时，比较每个词发出该声音的概率 P(X 声|W 词)，需要 GMM 模型和 HMM 模型。

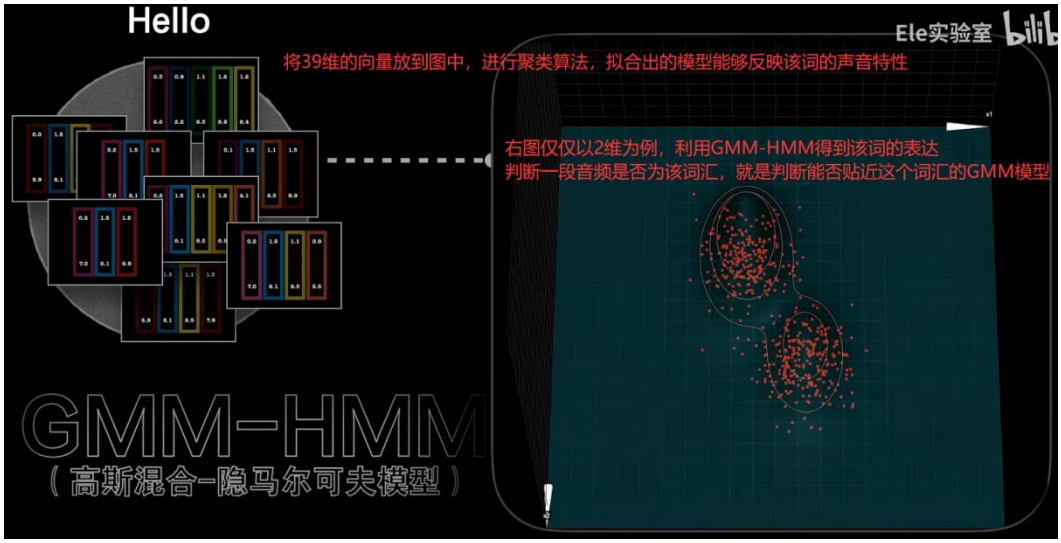
- x_j 表示第 j 个观测数据，$j = 1, 2, \ldots, N$
- K 是混合模型中子高斯模型的数量，$k = 1, 2, \ldots, K$
- α_k 是观测数据属于第 k 个子模型的概率，$\alpha_k \geq 0$，$\sum_{k=1}^{K} \alpha_k = 1$
- $\phi(x|\theta_k)$ 是第 k 个子模型的高斯分布密度函数，$\theta_k = (\mu_k, \sigma_k^2)$。其展开形式与上面介绍的单高斯模型相同
- γ_{jk} 表示第 j 个观测数据属于第 k 个子模型的概率

高斯混合模型的概率分布为：

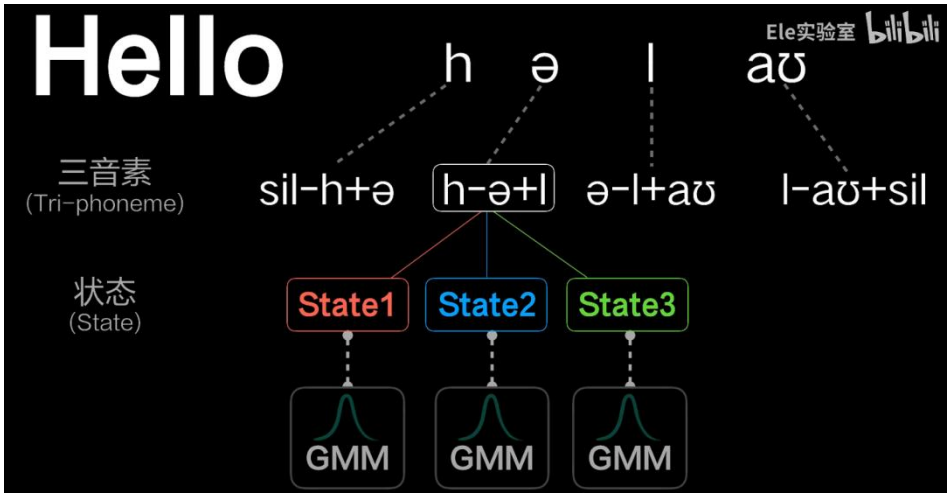
GMM高斯混合模型

$$P(x|\theta) = \sum_{k=1}^{K} \alpha_k \phi(x|\theta_k)$$

对于这个模型而言，参数 $\theta = (\tilde{\mu_k}, \tilde{\sigma_k}, \tilde{\alpha_k})$，也就是每个子模型的期望、方差（或协方差）、在混合模型中发生的概率。



找到声音最基本的状态，状态拼成音素，音素拼成字母，字母拼成单词。



又考虑到每个音素间也存在着前后的关系，这就要涉及 HMM 模型。

对于HMM模型，首先我们假设Q是所有可能的隐藏状态的集合，V是所有可能的观测状态的集合，即：

$$Q = \{q_1, q_2, \ldots, q_N\},\ V = \{v_1, v_2, \ldots v_M\}$$

HMM
隐马尔可夫模型

其中，N是可能的隐藏状态数，M是所有的可能的观察状态数。

对于一个长度为 T 的序列，I 对应的状态序列，O 是对应的观察序列，即：

$$I = \{i_1, i_2, \ldots, i_T\},\ O = \{o_1, o_2, \ldots o_T\}$$

其中，任意一个隐藏状态$i_t \in Q$，任意一个观察状态$o_t \in V$

HMM模型做了两个很重要的假设如下：

1） 齐次马尔科夫链假设。即任意时刻的隐藏状态只依赖于它前一个隐藏状态。当然这样假设有点极端，因为很多时候我们的某一个隐藏状态不仅仅只依赖于前一个隐藏状态，可能是前两个或者是前三个。但是这样假设的好处就是模型简单，便于求解。如果在时刻 t 的隐藏状态是 $i_t = q_i$，在时刻 $t+1$ 的隐藏状态是 $i_{t+1} = q_j$，则从时刻 t 到时刻 $t+1$ 的HMM状态转移概率 a_{ij} 可以表示为：

$$a_{ij} = P(i_{t+1} = q_j | i_t = q_i)$$

这样 a_{ij} 可以组成马尔科夫链的状态转移矩阵 A：

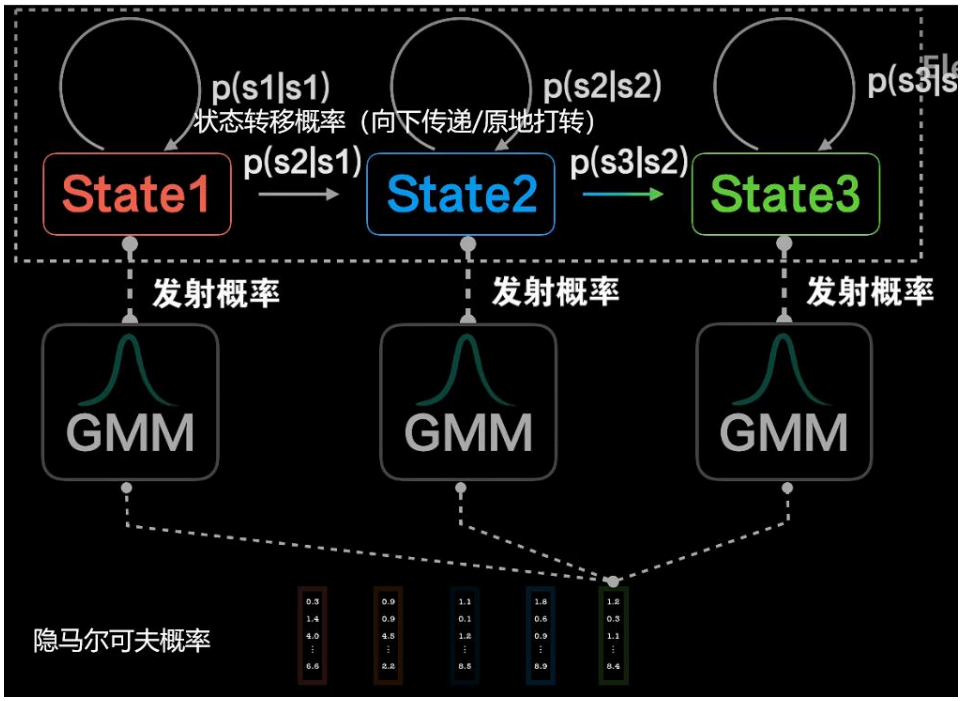
$$A = \left[a_{ij}\right]_{N \times N}$$

2） 观测独立性假设。即任意时刻的观察状态只仅仅依赖于当前时刻的隐藏状态。这也是一个为了简化模型的假设。如果在时刻 t 的隐藏状态是 $i_t = q_j$，而对应的观察状态为 $o_t = v_k$，则该时刻观察状态 v_k 在隐藏状态 q_j 下生成的概率 $b_j(k)$ 满足：

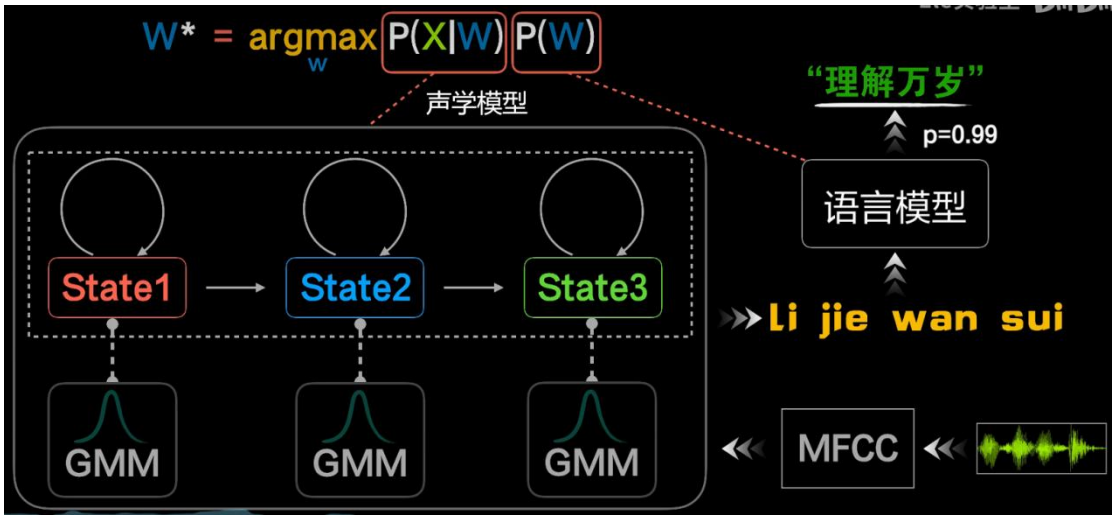
$$b_j(k) = P(o_t = v_k | i_t = q_j)$$

这样 $b_j(k)$ 可以组成观测状态生成的概率矩阵 B：

$$B = \left[b_j(k)\right]_{N \times M}$$



（三）**语言模型：**判断声学模型得出的一段话像不像话。



（四）深度学习的介入，语音识别技术彻底革新

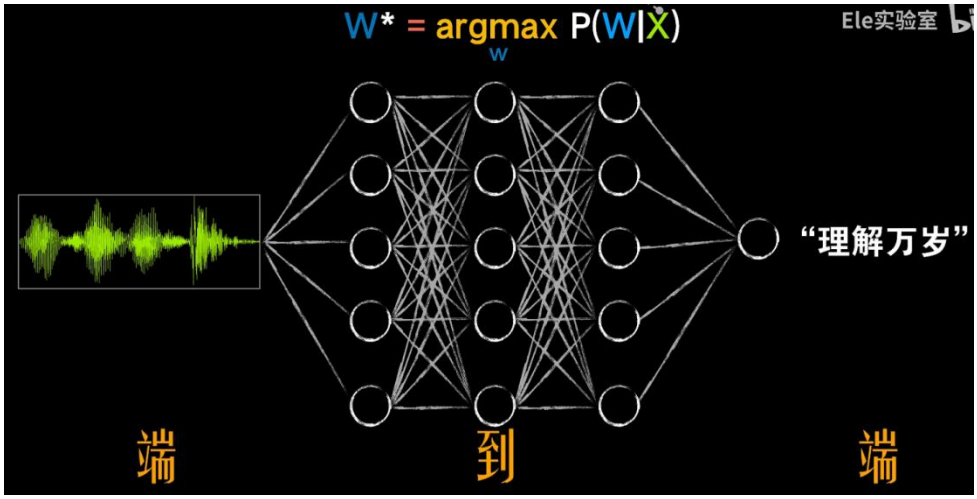
①MFCC 全部替换为神经网络找寻特征

②GMM 替换为神经网络让其自行拟合出待判别声音在每个已有词汇上的概率值。

由此 **GMM-HMM ——> DNN(深度神经网络)-HMM**

③语言模型 n-gram ——> 循环神经网络为主的（大）语言模型

④端到端：输入语音，直接识别输出文字



例如：基于循环神经网络（seq2seq 模型）的 LAS；CTC；RNN-T；

⑤语谱图 ~ 图像识别

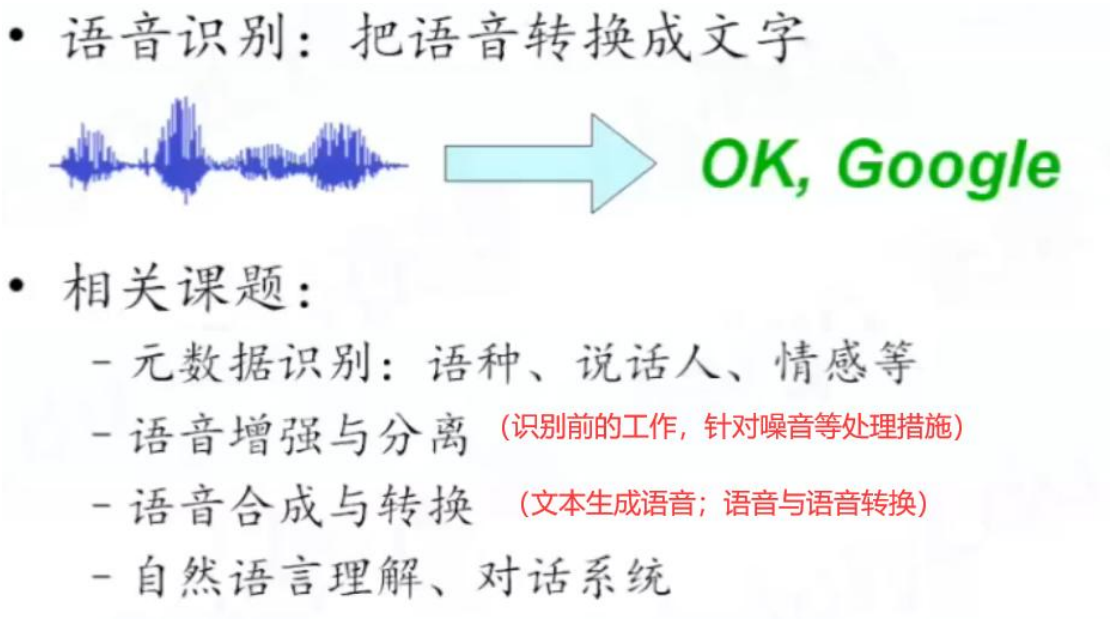
将语音（一段话或声音）识别问题转换看做图像识别问题，利用图像领域的技术。

例如：基于卷积神经网络的 DFCNN、SSA-IME

未完...

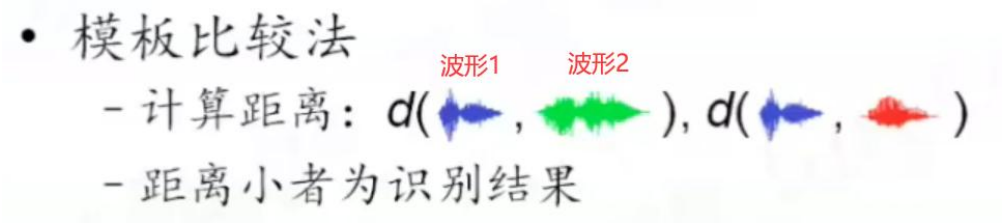
4.2 语音识别技术前世今生

语音识别技术的前世今生_哔哩哔哩_bilibili



前世：GMM+HMM

（1）孤立词识别



（2）特征提取

五. 语音信号处理（进阶）

5.1

六．语音识别技术（进阶）