重叠相加法和重叠保留法（ola ols）

主讲人：码大虾

ola和ols提出背景

信号处理中，通常处理的都是实时的音频信号，这样就使得我们在处理的过程中无法等信号接受完毕再做处理；一方面存储吃不消，另外一方面这也根本不能满足实际的要求。因此，在对长信号处理，提出了重叠相加法和重叠保留法两种处理方法，从而使得音频信号能够实时进行处理。另外，当两个信号长度相差较大的时候，往往也会考虑采用重叠相加法和重叠保留法来进行处理，这样大大提高了计算效率。

ola

- 设两个时间序列h(n)和x(n)，长度分别为N和N_1，两者相差很大，即N_1>>N；将x(n)进行分帧 $x_i(m)$ ，每帧长度和h(n)接近，然后将每帧和h(n)进行卷积，最后在相邻两帧之间时间重叠部分进行相加，这样就得到了x(n)和h(n)卷积结果；这就是重叠保留法的基本原理。具体如下：
- x(n)分帧为 $x_i(m)$,帧与帧之间不重叠，每帧长度为M，则有

$$x_i(m)=\begin{cases}x(n) & (i-1)M+1<=n<=iM\\ 0 & others\end{cases}$$

$$1<=m<=M,\quad i=1,2,...$$

ola

$$x(n)=\sum_{i=1}^{p}x_i(m),\quad p\text{为分帧后总帧数}$$

进一步分帧后，y(n)可以表示为：

$$y(n)=x(n)*h(n)=\sum_{i=1}^{p}x_i(n)*h(n)$$

$$\Rightarrow y(n)=\sum_{i=1}^{p}y_i(n)$$

通过卷积学习可知,y(n)的长度为N + M - 1，而 $x_i(m)$ 的长度为M，所以相邻两帧 $y_i(n)$ 之间有N-1长度的数据是时间上重叠的，重叠

ola

- 部分相加，与不重叠的部分共同构成输出。

ola

- 例1. h=[1,-2,-3]，x=[2,-3,4,5,-6,7,8,-9,-10,11,-12,-13,-14]，取L=5对x进行分段，利用重叠保留法求解y=x*h。
- y=[2,-7,4,6,-28,**4,12**,-46,-16,58,**-4,-22**,48,67,42]

$$x_0(n)=[2,-3,4,5,-6] \qquad y_0(n)=x_0(n)*h(n)=[2,-7,4,6,-28,-3,18]$$

$$x_1(n)=[7,8,-9,-10,11] \qquad y_1(n)=x_1(n)*h(n)=[7,-6,-46,-16,58,8,-33]$$

$$x_2(n)=[-12,-13,-14,0,0] \qquad y_2(n)=x_2(n)*h(n)=[-12,11,48,67,42,0,0]$$

ols

- 重叠保留法和重叠相加法类似，都是用来处理两个长度相差很大的时间序列的或者说用于处理实时音频数据处理。对于两个时间序列h(n),x(n),其长度分别为N,N_1,且N_1>>N。将x(n)进行分帧为 $x_i(m)$ 且帧与帧之间没有重叠，帧长和h(n)接近，然后将每帧 $x_i(m)$ 和h(n)进行卷积得到输出结果。具体过程如下：
  这里假设h(n)为时不变的，对h(n)补零：

$$\tilde{h}(m)=\begin{cases}h(n) & 1<=n,m<=N\\ 0 & N+1<=m<=N+M-1\end{cases}$$

ols

- $x_i(m)$的处理方法不同于重叠相加法，每帧长为N+M-1，并要求每帧分帧的最后一个数据点都在$_i$M处；对于第一帧由于数据长度只有M，所以向前补N-1个零，使得第一帧长度为N+M-1；后续的帧可以延用前一帧的N-1个数据点，从而达到帧长为N+M-1。

ols

- 例1. h=[1,-2,-3]，x=[2,-3,4,5,-6,7,8,-9,-10,11,-12,-13,-14]，取L=5对x进行分段，利用重叠保留法求解y=x*h。
- y=[2,-7,4,6,-28,4,12,-46,-16,58,-4,-22,48,67,42]

$$x_0(n)=[\underline{0,0},2,-3,4]$$
$$x_1(n)=[\underline{-3,-4},5,-6,7]$$
$$x_2(n)=[\underline{-6,7},8,-9,-10]$$
$$x_3(n)=[\underline{-9,-10},11,-12,-13]$$
$$x_4(n)=[-12,-13,-14,0,0]$$

$$y_0(n)=[\underline{1,-12},2,-7,4]$$
$$y_1(n)=[\underline{1,-11},6,-28,4]$$
$$y_2(n)=[\underline{41,49},12,-46,-16]$$
$$y_3(n)=[\underline{53,47},58,-4,-22]$$
$$y_4(n)=[\underline{-12,11},48,67,42]$$

程序实现

- ola:

```
h = [1,-2,-3];
x0 = [2,-3,4,5,-6];
x1 = [7,8,-9,-10,11];
x2 = [-12,-13,-14];
```