



## 第一次作业（第二章）讲评



主讲人 俞帆



- 第一部分：作业情况概述
- 第二部分：作业讲评

- 总体都能正确完成，实现细节上会有不同
- 实际应用中，特征提取应该注意训练测试流程保持一致，实现细节上的差异不会有太大影响
- 鼓励参考一些工具的特征提取流程，看看与自己实现的有什么不同

- 第一部分：作业情况概述
- 第二部分：作业讲评

- 滤波器设计：
  - 滤波器刻度设计

```
low_mel_freq = 0
high_mel_freq = 2595 * np.log10(1 + (fs / 2) / 700) # 转换到mel尺度
mel_filters_points = np.linspace(low_mel_freq, high_mel_freq, num_filter + 2) #mel空间中线性取点
freq_filters_pints = (700 * (np.power(10., (mel_filters_points / 2595)) - 1)) #转回线性谱
freq_bin = np.floor(freq_filters_pints / (fs / 2) * (fft_len / 2 + 1)) #把原本的频率对应值缩放到FFT窗长上
```

## ● 滤波器设计:

### ● 滤波器设计

```
feats = np.zeros((int(fft_len / 2 + 1), num_filter))
for m in range(1, num_filter + 1):
    bin_low = int(freq_bin[m - 1]) #每一个滤波器的起始点
    bin_medium = int(freq_bin[m]) #每一个滤波器的中点（最高点）
    bin_high = int(freq_bin[m + 1]) #每一个滤波器的终点
    for k in range(bin_low, bin_medium): #上升部分: 0-1
        feats[k, m - 1] = (k - freq_bin[m - 1]) / (freq_bin[m] - freq_bin[m - 1])
    for k in range(bin_medium, bin_high): #下降部分: 1-0
        feats[k, m - 1] = (freq_bin[m + 1] - k) / (freq_bin[m + 1] - freq_bin[m])
```

## ● 滤波器设计:

### ● 滤波、取对数

- 输入可以取平方（计算能量）
- 输入的幅值谱可以转为功率谱再滤波
- 取log部分可以计算声压级

```
feats = np.dot(spectrum, feats) # (356, 257),(257, 23)
feats = 20 * np.log10(feats) # log
```

## ● 计算MFCC:

- DCT注意输出维数
- 可以计算能量,  $\Delta$  以及double  $\Delta$

**感谢各位聆听 !**  
Thanks for Listening

