

**实验报告**

**实 验（二）**

题 目 DPCM编解码实验

专 业 视听觉信息处理

学　　 号 1180300107

班　　 级 1803105

学 生 胡云帆

指 导 教 师 郑铁然

实 验 地 点 格物207

实 验 日 期 2020.12. 23

**计算机科学与技术学院**

# 一、 8比特DPCM编解码算法

## 简述算法内容

对输入的语音，如果对相邻两项做差，进行量化，可以得到编码内容。但是，这样会造成噪音累积的效果，这是不能接受的，为了解决这个问题，在输入端做一个解码器，对信号进行编码后，再进行解码，所求差是原待编码量与前一项解码后的量，这样做即不会累积误差。具体实现公式如下：

1.2 解码信号的信噪比

其中，是解码后的信号,是原信号

通过计算信噪比，即可比较压缩的损失。

# 二、4比特DPCM编解码算法

## 2.1 你所采用的量化因子

a = 591

## 2.2拷贝你的算法，加上适当的注释说明

def factor\_DPCM(wave\_data:np.ndarray,a:float,bits:int=4):

x = wave\_data

x\_hat = np.zeros(wave\_data.shape)

d = np.zeros(wave\_data.shape)

d[0] = x[0] #第一项直接拷贝

x\_hat[0] = wave\_data[0]

c = np.zeros(wave\_data.shape)

c[0] = factor\_encode(d[0],a,bits) #第一项进行编码

for i in range(1, wave\_data.shape[0]):

d[i] = x[i] - x\_hat[i - 1] #与前一项编码后解码的结果做差

c[i] = factor\_encode(d[i],a,bits)

x\_hat[i] = x\_hat[i - 1] + (c[i]-1)\*a #直接解码本项留作后用

return c

def factor\_encode(num:int,a:float=5,bits=4):

x = 2 \*\* (bits-1)

if num > (x-1)\*a:

return (x-1)

elif num < -x \* a:

return -x

else:

return num//a + 1

def factor\_decode\_DPCM(c:np.ndarray,a:int,bits:int=4):

x = 2 \*\* (bits - 1)

x\_hat = np.zeros(c.shape)

x\_hat[0] = c[0]

for i in range(1, c.shape[0]):

x\_hat[i] = x\_hat[i - 1] + (c[i]-1)\*a #解码公式

return x\_hat

采取了量化因子法。

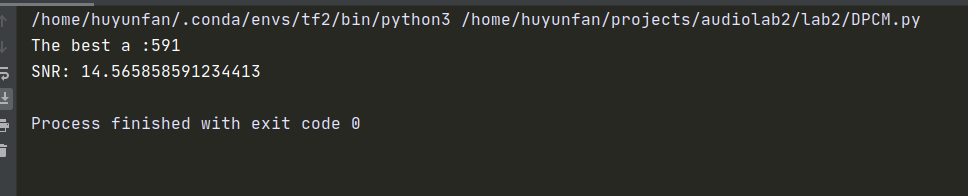
编码算法如下所示：



解码算法如下所示：



2.3 解码信号的信噪比



# 三、改进策略

## 3.1 你提出了什么样的改进策略，效果如何

首先随意选择a的值得到效果较差。后来发现随着a值的变化，SNR函数凸，即存在一个极值。由于仅对一个文件进行处理，这个极值容易找到。我仅仅在1和1000中进行暴力枚举，即发现选择591对4bit和8bit均是SNR最大的，但是处理后二者的SNR并不相同，这是显然的。得到8bit时SNR是17左右，4bit时SNR是14.5左右，理论上是比较好的结果。听过声音效果后，发现与原声音区别不大，仅仅存在很小的噪音，语音信息基本未损失，是可以接受的。

# 四、 简述你对量化误差的理解

## 4.1 什么是量化误差？

量化误差是对信号进行量化产生的误差。该误差反应了量化结果和被量化量的差值。如在本例量化过程中，比例的缩放会造成求余的过程产生误差，较大值的舍入会造成误差。

4.2 为什会编码器中会有一个解码器

这是为了避免误差的累计。如果不使用解码器，直接做差，即

已知解码的过程中

可以得到以下推导

这样会造成误差的累计，产生很大的误差。

使用解码器，并不进行直接做差，而是

代入上式即可直接得到

这样即可避免误差的累计。

# 五、 总结

## 5.1 请总结本次实验的收获

对语音编码有了新的理解。对DPCM有了更深刻的理解，训练了编码能力

## 5.2 请给出对本次实验内容的建议

由于学校受限，建议将PPT中的示例代码片段更换为同学们的实现中实现得较为优秀的python代码或其他语言代码