

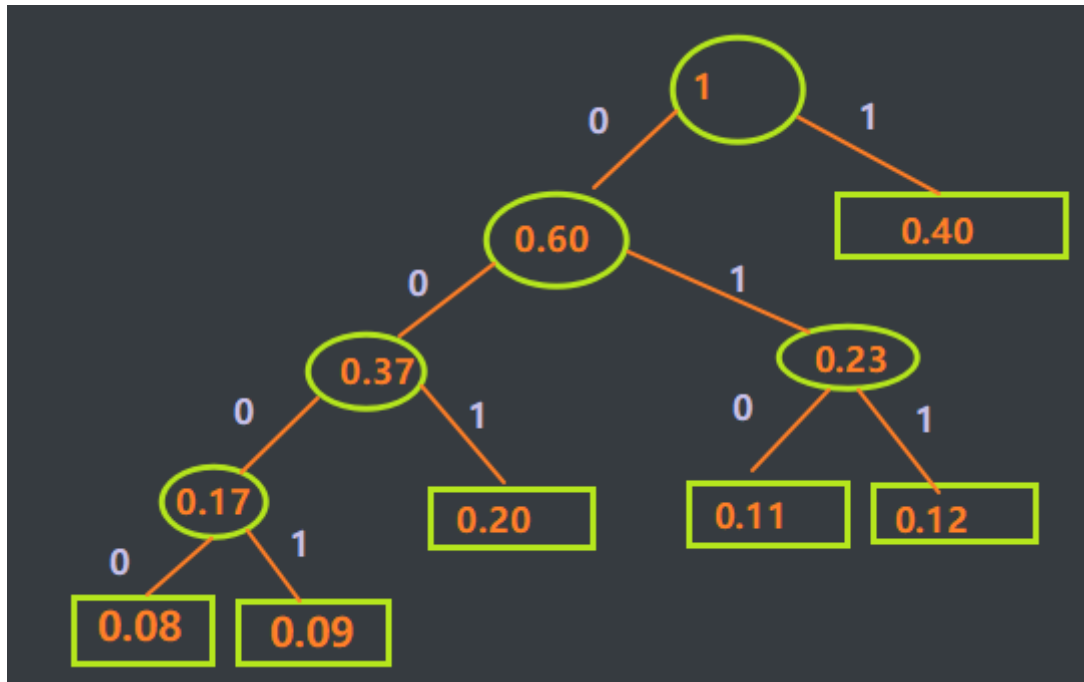
# 第8章作业

计创18-8-连月菡

## 1. 对信源符号（概率） $x_1(0.4), x_2(0.20), x_3(0.12),$

$x_4(0.11), x_5(0.09), x_6(0.08)$  进行哈夫曼编码。

(对本PPT中的例子中的霍夫曼编码进行验证)



## 2. 对信源符号（概率） $x_1(0.4), x_2(0.30), x_3(0.1),$

$x_4(0.1), x_5(0.06), x_6(0.04)$  进行香农编码。（借助MATLAB实现）

```
1 % 香农编码matlab实现
2 pa = [0.4;0.3;0.1;0.1;0.06;0.04]; %输入概率
3 k=length(pa); %计算信源符号个数
4 if min(pa)<0||max(pa)>1 %判断信源概率值是否介于0到1之间
5     % disp(['信源分布pa(x)=[',num2str(pa),']']);
6     disp('概率值必须介于0到1之间，请重新输入信源分布');
7     return
8 elseif sum(pa)~=1 %判断信源累加和是否为1
9     % disp(['信源分布pa(x)=[',num2str(pa),']']);
10    disp('概率累加和必须等于1，请重新输入信源分布');
11    return
12 else
13     for i=1:k-1 %for循环进行降序排列
14         for n=i+1:k
15             if (pa(i)<pa(n))
16                 t=pa(i);
17                 pa(i)=pa(n);
18                 pa(n)=t;
19             end
20         end
21     end
```

```

22 end
23 disp('信源分布概率从大到小为: '),disp(pa)
24 y=0;%给y赋初值,用来求概率和
25 f=0;%给f赋初值,用来得到子程序最大循环次数
26 s=zeros(k,1); %对求和结果进行矩阵初始化
27 b=zeros(k,1); %对编码位数矩阵初始化
28 w=zeros(k,1); %对二进制矩阵初始化
29 for m=1:k %进行香农编码
30     s(m)=y;
31     y=y+pa(m);
32     b(m)=ceil(-log2(pa(m)));%求得的自信息量向上取整,得到码字长度
33     z=zeros(b(m),1); %对码字矩阵初始化
34     x=s(m);
35     f=max(b(m)); %把码字最大长度赋给f,用于进行十进制转二进制
36     w=dtob(x,f); %调用子程序将十进制转换为二进制
37     for r=1:b(m)
38         z(r)=w(r);
39     end
40     disp('输出结果为: ');
41     disp('初始概率'),disp(pa(m))
42     disp('求和结果'),disp(s(m))
43     disp('编码位数'),disp(b(m))
44     disp('最终编码'),disp(z')
45
46 end
47 sum0=0;
48 sum1=0;
49 for i=1:k %使用for循环进行信息熵、平均码长求解
50     a(i)=-log2(pa(i)); %a(i)表示单个信源的自信息量
51     K(i)=ceil(a(i)); %K(i)表示对自信息量向上取整
52     R(i)=pa(i)*K(i);
53     sum0=sum0+R(i); %求平均码长
54     c(i)=a(i)*pa(i);
55     sum1=sum1+c(i); %信息熵
56 end
57 K1=sum0;
58 H=sum1;
59 Y=H/K1; %用Y来表示编码效率
60 disp(['信息熵H(X)=',num2str(H),'(bit/sign)']);
61 disp(['平均码长k=',num2str(K1),'(bit/sign)']);
62
63 disp(['编码效率=',num2str(Y)]);
64
65 function y=dtob(x,f)
66 for i=1:f
67     temp=x.*2;
68     if(temp<1)
69         y(i)=0;
70         x=temp;
71     else
72         x=temp-1;
73         y(i)=1;
74     end
75 end
76 end

```

1 输出结果为:

初始概率

0.0600

求和结果

0.9000

编码位数

5

最终编码

1 1 1 0 0

0 输出结果为:

初始概率

0.0400

求和结果

0.9600

编码位数

5

最终编码

1 1 1 1 0

1 信息熵 $H(X)=2.1435$  (bit/sign)

平均码长 $K=2.7$  (bit/sign)

编码效率=0.7939

输出结果为:

初始概率

0.1000

求和结果

0.7000

编码位数

4

最终编码

1 0 1 1

输出结果为:

初始概率

0.1000

求和结果

0.8000

编码位数

4

最终编码

1 1 0 0

>> Untitled

信源分布概率从大到小为:

0.4000

0.3000

0.1000

0.1000

0.0600

0.0400

输出结果为:

初始概率

0.4000

求和结果

0

编码位数

2

最终编码

0 0

输出结果为: |

初始概率

0.3000

求和结果

0.4000

编码位数

2

最终编码

0 1

信源符号	概率分布	累计概率	位数	编码
X1	0.40	0	2	00
X2	0.30	0.40	3	01
X3	0.10	0.70	4	1011
X4	0.10	0.80	4	1100
X5	0.06	0.90	5	11100
X6	0.04	0.96	5	11110

3.信源符号（概率）x1(0.1),x2(0.2),x3(0.3),x4(0.4). 输入符号串为:

“x4 x3 x4 x2 x3”，对该符号串进行算术编码

信源符号	区间	区间长
x4	[0.6,1]	0.4
x3	[0.72,0.84]	0.12
x4	[0.792,0.84]	0.048
x2	[0.7968,0.8064]	0.0092
x3	[0.79968,0.80256]	0.00288

取0.80175，对应二进制为0.1100110101。

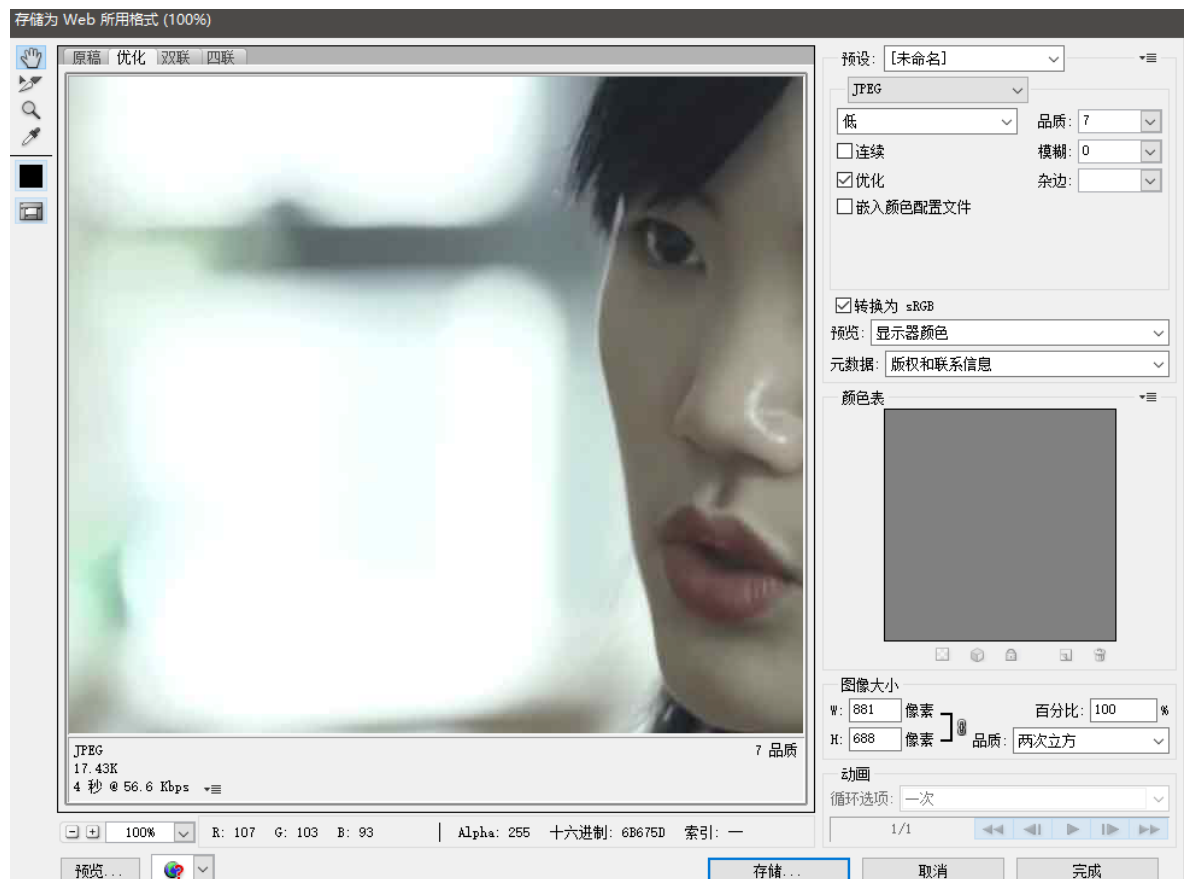
## 4.掌握行程编码的方法













行程编码的基本原理是在给定的数据图像中寻找连续的重复数值，然后用两个字符取代这些连续值。

在此方式下每两个字节组成一个信息单元。第一个字节给出其后面相连的像素的个数。第二个字节给出这些像素使用的颜色索引表中的索引。例如：信息单元03 04，03表示其后的像素个数是3个，04表示这些像素使用的是颜色索引表中的第五项的值。压缩数据展开后就是04 04 04。同理04 05 可以展开为05 05 05 05。信息单元的第一个字节也可以是00，这种情况下信息单元并不表示数据单元，而是表示一些特殊的含义。这些含义通常由信息单元的第二个字节的值来描述。

## 5.在 Photoshop软件中以JPEG格式储存时，提供11级压缩级别，以0—10级表示。其中0级压缩比最高，图像品质最差。

对一幅图像分别用0-10级进行压缩，比较一下采用哪一级压缩可以达到存储空间与图像质量兼得的最佳比例。



	1	2020/5/18 14:59	JPG 文件	17 KB
	1	2020/5/14 21:53	Markdown File	1 KB
	1原图	2020/5/18 14:54	PNG 文件	348 KB
	2	2020/5/18 14:58	JPG 文件	17 KB
	3	2020/5/18 14:58	JPG 文件	17 KB
	4	2020/5/18 14:58	JPG 文件	18 KB
	5	2020/5/18 14:58	JPG 文件	18 KB
	6	2020/5/18 14:57	JPG 文件	18 KB
	7	2020/5/18 14:57	JPG 文件	19 KB
	8	2020/5/18 14:56	JPG 文件	19 KB
	9	2020/5/18 14:56	JPG 文件	20 KB
	10	2020/5/18 14:56	JPG 文件	20 KB

我认为采用6级压缩比较好















