

第三章 模型驱动的数据再表达学习方法

3.1 引言

人类提升对世界的认识能力的方法就是从现实世界中发现规律，从认识论的角度来说就是从感性认识到理性认识。那么规律如何描述呢？从自然科学的角度来看，人类描述自然规律的主要方法是用数学公式的方法，将自然规律用一个数学公式（或者类似的数学工具）表达，数学公式就是模型（model）或者叫做模式（pattern）。

发现数学公式或类似的数学工具的方式有两类：

（1）所谓的基于数据驱动的数学建模方式，也就是从大量的实际发生的数据中发现数量之间关系并且用数学公式的方式体现出来。这类方法需要实际数据，那么第一步就是从现实中采集数据，然后分析数据的性质，发现规律，建立数学模型，进而描述数据的特性。

（2）所谓基于模型驱动的方式，也就是围绕着一般的函数类（一般的信号）和一般的算子（变换），把注意力集中在这些一般函数（信号）的定量性质上，并且研究当不同的算子作用于它们时，这些定量性质是怎样变化的。

从理论上讲，只要有足够代表性的样本（数据），就可以运用数学的方法找到一个或者一组模型的组合使得它和真实的情况非常接近。注意这个方法的前提是具有“**足够代表性的数据**”。这种方法称为“**数据驱动方法**”。之所以被称为数据驱动方法是因为它是现有大量的数据，而不是预设模型，然后用很多简单的模型来契合数据。虽然通过这种方法找到的模型可能和真实模型存在一定的偏差，但是在误差允许的范围内，单从结果上和精确的模型是等效的。当然从数据出发也有可能得到真实的模型，但是这并非数据驱动方法的目标。可以看出来数据驱动的方法目标就是近似替代，它甚至不是为了追求真实。数据驱动方法的意义在于，**当我们对一个问题暂时不能用简单而准确（一般真实的原理都是简单而准确的）的方法解决时，我们可以根据以往的历史数据，构造出近似的模型来逼近真实情况，这实际上是用计算量和数据量来换取研究时间。**得到的模型虽然和真实情况有偏差但是足以指导实践。而且数据驱动方法有一个特别大的优势，就是能够最大程度的得益于计算机技术的进步。如此一来人类发现真理的速度会越来越快的。

数据驱动的方式一般分两步：

第一步：确定数学模型（狭义模型）；

第二步：确定模型中的参数。

如第二章我们介绍的机器学习中数据驱动的前端学习方法，特别是逐层学习方法，都是分成这两个步骤。它的使用前提是，有大量的具有代表性的数据，仅仅有大量数据还不够，数据还必须具备代表性。

数学理论方法蕴含着处理智能问题的基本思想与方法，也是理解复杂算法的必备要素。现在的人工智能技术，包括机器学习技术，归根到底都建立在数学模型之上。本章我们介绍模型驱动的数据再表达方法，首先以调和与分析为工具，介绍模型驱动的数据再表达方法，然后介绍两种具体的模型驱动的数据稀疏再表达方法。

定义 3.1: 不需要任何实际训练样本，仅通过模型就可以得到的揭示输入数据潜在变化因素的数据再表达方法称为模型驱动的数据再表达方法。

调和分析方法的一个特点是用这种方法在分析一般函数（信号）时，多数是通过一些局部性质组合得到函数的整体。例如，在分析一个函数 $f(t)$ 时，我们把它分解为一个和： $f(t) = f_1(t) + \cdots + f_N(t)$ ，而这里的每个函数都是“局部化”了的，即它的支集（就是使得 $f_i(t) \neq 0$ 的点 t 的集合的闭包）的直径都很小。这种分解叫做空间变量的局部化。在用 **Fourier** 变换把 $f(t)$ 变换成 $\hat{f}(\omega)$ 后，也可以对频率变量 ω 进行局部化。对 $f(t)$ 分割开以后，就可以对这些小块分别进行估计（即再表达），以后再把它们重组合起来，就是对原始函数（信号）的一种再表达。实现这种“分而治之”的再表达战略的理由是：一个典型的函数 $f(t)$ 倾向于同时具有若干不同的特点。直观上，它们在有些地方可以是“有尖峰”的、有些地方可以是“有低谷”的、不连续的、“具有高频”的，而在其它地方则是“光滑的”或者“具有低频”的。同时抽取这些特征，同时处理所有这些特点，即一次把所有的特征都抽取出来是困难的。那么，一个很好的选择就是分解。通过分解，把这些特点逐步分离出来，使得每一个成分都只有一个突出的可能导致困难的特点：尖峰（低谷）出现在一个 $f_i(t)$ 里面，高频则出现在另外一个 $f_i(t)$ 里面，如此等等。再把各个成分的估计（再表达）重新组合起来，就得到原始信号的一种逼近（再表达）。例如 **Fourier** 变换和小波变换都是典型的具体实现方式。这种函数的再表达，不是依据某些实际数据获得，而是针对某一类抽象的函数空间里的所有函数，如 $L^2(\mathbb{R})$ 。下面首先介绍基于熵编码的数据再表达方法，然后介绍基于离散余弦变换（**DCT**）的数字图像数据的稀疏再表达方法，再介绍基于多分辨率分析的一般函数分解方法以及基于小波变换的数字图像数据的逐层再表达方法。这都可以看成模型驱动的数据逐层再表达学习方法。

3.2 基于熵编码的数据再表达方法

在一段有意义的实际数据中，都会有许多重复片段，这些重复使得数据的样本之间具有相关性。数据再表达的一个重要目标就是去除这些相关性，其直接应用是再表达后的数据可以减少存储空间。用计算机处理数据时，数据的表现方法通常是存储体的最小单位表示，常见的方式是以一个字节（8 个位）表示一个字符，因此最多可以表示 256 种字符，即计算机表示数据时都是用这 256 个字符表示。以有限的符号来表达一段又若干样本组成的数据时，必然会产生许多重复，这些重复也表示数据之间潜在的相关性。

数据所包含的“信息量”是数据的一个重要特性，许多特征都与“信息量”有关。信息量的另外一个称谓是信息熵（乱度，Entropy），通常信息熵所表示的意义就是混乱的程度，对于各种数据而言，若混乱程度越高，则数据所能够表达的意义就越丰富。反之，若数据的重复性很高，或者该数据中的样本之间遵循某种规则，则该数据所包含的意义就不多。假设某个字符（事件）出现的概率为 p ，

则该字符（事件）的信息量（熵）定义为 $I = -\log p$ 。另一个与信息熵有关的决定数据中所包含的信息量多少的一个概念是多余信息量（Redundancy），或者称为冗余量。这种冗余量实际就是数据相关性的一种表示方法，数据再表达的一个重要目标是尽可能将这种冗余量去除。这种数据再表达的两个代表性方法是基于 Huffman 编码和基于算术编码的数据再表达方法，下面分别介绍。

3.2.1 基于 Huffman 编码的数据再表达

基于 Huffman 编码的数据再表达过程如下。

1. 信源符号的 Huffman 码字的生成：

第 1 步：将表达数据的所有信源符号出现的概率按从低到高排序；

第 2 步：依次将出现概率最低的两个信源符号结合形成新的合成符号，此新符号称为节点，合成符号出现的概率为合成这个新符号的两个符号的概率之和。这个新的节点也可和尚未结合成节点的符号相结合，形成另一个新的节点，已经结合为节点的符号或者用来结合成新节点的节点不再考虑；

第 3 步：反复进行第 2 步，至到最后结合为一个节点为止，得到一棵树，其中数中的叶子为表达数据的所有符号，每个节点内含该节点两个子节点（或者树叶）的概率之和；

第 4 步：依次由树根出发，前进到每一个叶子（每一个信源符号），每遇到一个节点，左边的分支分配一个 0，右边的分支分配一个 1，把由树根到叶子的路径中所遇到的 0 和 1 串联起来，得到该符号的码字；

2. 基于 Huffman 编码的数据再表达

对于输入数据，依次将出现的信源字符按上述产生的 Huffman 码字进行再表达，即可得到基于 Huffman 编码的数据再表达形式。

例如：假设信源符号为 a_1, a_2, \dots, a_9 ，对应的出现概率为：

符号	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
概率	$p_1 = 0.1$	$p_2 = 0.05$	$p_3 = 0.3$	$p_4 = 0.15$	$p_5 = 0.01$	$p_6 = 0.1$	$p_7 = 0.2$	$p_8 = 0.09$

	码字	信源符号	概率
	111111	a_5	0.01
	111111	a_2	0.05
	111111	a_8	0.09
	111111	a_1	0.10
	111111	a_6	0.10
	111111	a_4	0.15
	111111	a_7	0.20
	111111	a_3	0.30

图 3.1 信源符号的 Huffman 码字的生成

给定一个数据，如为 $a_2 a_9 a_1 a_3 a_9 a_3 a_1 a_3 a_4 a_9 a_5 a_6 a_7 a_3$ ，则该数据的再表达为：

3.2.2 基于算术编码的数据再表达

信源符号的 Huffman 码字的生成中，一个符号至少需要一个位(bit)来表示，这通常不是最优的。可以想象，当一个信源字符出现的概率非常大时，完全可以用少于一个位表示。算术编码就很好的解决了这个问题。算术编码用一个单个的浮点数值来表达一个数据，无论该数据有多长。对于较长的复杂数据，输出的数值就需要更多的位数。理论上，不管输入数据有多长，只要浮点数值位数足够多，都可以对该数据进行再表达。

基于算术编码的数据再表达过程中产生的再表达形式是一个 $[0,1)$ 中的数值，这个数值可以唯一建立这种数据再表达。为了构造这样的数据在表达，符号需要被赋以一系列概率值，这些概率值可以通过训练数据统计得到，也可以事先根据概率分布得到。例如：假设输入数据为 *illuminate*，有如下的概率分布：

表 3.1 信源符号及其出现的概率

字符	a	e	i	l	m	n	t	u
概率	1/10	1/10	2/10	2/10	1/10	1/10	1/10	1/10

概率确定后，每个符号需要赋以 $[0,1)$ 中的一个数值范围：

表 3.2 信源符号及其概率范围

字符	概率	范围
a	1/10	$0 \leq r < 0.1$
e	1/10	$0.1 \leq r < 0.2$
i	2/10	$0.2 \leq r < 0.4$
l	2/10	$0.4 \leq r < 0.6$
m	1/10	$0.6 \leq r < 0.7$
n	1/10	$0.7 \leq r < 0.8$
t	1/10	$0.8 \leq r < 0.9$
u	1/10	$0.9 \leq r < 1$

每个符号都赋值成 $[0,1)$ 中的一个与其出现的概率相关的数值范围。数据再表达的最高有效部分属于第一个符号“i”，为了正确表达这个数据，其数据再表达的数值应该在 $[0.2,0.4)$ 中，即数据 **illuminate** 的数据再表达的最终值在 $[0.2,0.4)$ 中。在表达剩余部分中，每个新符号将进一步限制输出值的可能范围。下一个符号为“l”，拥有子范围 $[0.2,0.4)$ 中的范围 $[0.4,0.6)$ ，即数据再表达的最终值在范围 $[0.2,0.4)$ 中的 40%到 60%之间，即在 $0.20 + (0.40 - 0.20) \times 0.4 = 0.28$ 和 $0.20 + (0.40 - 0.20) \times 0.6 = 0.32$ 之间。因此，数据再表达的范围值为 $[0.28,0.32)$ 。依次类推：

表 3.3 数据的编码取值范围

新字符	低值	高值
	0	1
i	0.2	0.4
l	0.28	0.32
l	0.296	0.304
u	0.2996	0.3000
m	0.29984	0.29988
i	0.299848	0.299856
n	0.2998536	0.2998544
a	0.29985360	0.29985440
t	0.299853664	0.299853672

e	0.2998536641	0.2998536642
---	--------------	--------------

在最后一个符号“e”的低值 0.2998536641 和高值 0.2998536642 之间任意取一个实数，如 0.29985366415，即为数据 *illuminate* 的数据再表达的最终值。译码过程非常简单，首先 $0.29985366415 \in [0.2, 0.4)$ ，所以第一个符号为“i”，减掉符号为“i”的下界并除以“i”的宽度得到数值 0.4992683207，即得下一个符号为“l”，依次类推，最后可以得到数据 *illuminate*。理论上说，只要支集长度足够长，任意厚度的一本小说，或者一本字典上的所有信息，都可以用一个实数进行再表达。

3.3 基于离散余弦变换和熵编码的图像数据逐层再表达方法

3.2.1 可分离变换

二维离散 Fourier 变换和逆变换的通用公式为：

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) g(x, y, u, v) \quad (3.1)$$

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) h(x, y, u, v) \quad (3.2)$$

$(x, u=0, 1, \dots, M-1; y, v=0, 1, \dots, N-1)$ ； $g(x, y, u, v)$, $h(x, y, u, v)$ 分别称为正向变换核和反向变换核。如果满足：

$$g(x, y, u, v) = g_1(x, u) g_2(y, v) \quad (3.3)$$

$$h(x, y, u, v) = h_1(x, u) h_2(y, v) \quad (3.4)$$

则称正、反变换核是可分离的。进一步，如果 g_1 和 g_2 ， h_1 和 h_2 在函数形式上一样，则称该变换核是对称的。

3.2.2 图像变换的矩阵表示

数字灰度图像一般指 256 级（8 位）数字图像，将颜色由黑色到白色分成了 256 级，它实际上就是一个矩阵，矩阵的维数代表图像的大小，矩阵中的元素代表像素值，取值为 0 到 255，其中 0 代表该像素为纯黑色，255 代表该像素为纯白色。下面介绍的此类图像数据的稀疏表达方法是静态图像压缩国家标准 JPEG 中的核心技术。

给定一幅 $M \times N$ 大小的图像，记为 $(f(x, y))_{M \times N}$ ，其中 $0 \leq f(x, y) \leq 255$ ， $(x=0, 1, \dots, M-1; y=1, 2, \dots, N-1)$ 。与一维的有限长离散非周期信号存在傅里叶

变换 (DFT) 一样, 图像作为一个二维离散信号同样存在着二维离散变换, 如二维离散信号的离散 Fourier 变换 (DFT)、离散余弦变换 (DCT)、离散小波变换 (DWT) 等。通常为了分析、推导方便, 可将可分离变换写成矩阵的形式:

$$F = PfQ$$

$$f = P^{-1}FQ^{-1}$$

其中, F, f 是 $M \times N$ 的矩阵; P 是 $M \times M$ 矩阵; Q 是 $N \times N$ 矩阵。

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} P(x, u) f(x, y) Q(y, v) \quad (3.5)$$

其中 $(u=0, 1, \dots, M-1; v=1, 2, \dots, N-1)$ 。

对二维离散傅立叶变换, 则有:

$$P(x, u) = g_1(x, u) = e^{-j2\pi ux/M} \quad (3.6)$$

$$P(y, v) = g_2(y, v) = e^{-j2\pi vy/N} \quad (3.7)$$

实际上, 除了 DFT 变换之外, 还采用许多其他的可分离的正交变换。例如: 离散余弦变换、沃尔什-哈达玛变换、K-L 变换等。

3.2.3 离散余弦变换

离散余弦变换 (Discrete Cosine Transform, DCT) 是可分离的变换, 其变换核为余弦函数。DCT 除了具有一般的正交变换性质外, 它的变换阵的基向量能很好地描述人类语音信号和图像信号的相关特征。因此, 在对语音信号、图像信号的变换中, DCT 被认为是一种准最佳变换, 是对原始信号的一种很好的再表达的工具。

1. 一维离散余弦变换

对于时域中的点序列 $\{f(x) | x = 0, 1, \dots, N-1\}$ 为离散的信号列, 则其一维离散余弦变换为:

$$F(0) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} f(x) \quad (3.8)$$

$$F(u) = \sqrt{\frac{2}{N}} \sum_{x=0}^{N-1} f(x) \cos \frac{2(x+1)u\pi}{2N}$$

其中 $F(u)$ 是第 u 个余弦变换系数, u 是广义频率变量, $u = 0, 1, \dots, N-1$ 。

逆变换为:

$$f(x) = \sqrt{\frac{1}{N}} F(0) + \sqrt{\frac{2}{N}} \sum_{u=0}^{N-1} F(u) \cos \frac{(2x+1)u\pi}{2N} \quad (3.9)$$

上面两个式子 (3.8) 和 (3.9) 构成了一维离散余弦变换对。

2. 二维离散余弦变换

正变换为:

$$\begin{aligned}
 F(0,0) &= \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \\
 F(0,v) &= \frac{\sqrt{2}}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cdot \cos \frac{(2y+1)v\pi}{2N} \\
 F(u,0) &= \frac{\sqrt{2}}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos \frac{(2x+1)u\pi}{2N} \\
 F(u,v) &= \frac{2}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos \frac{(2x+1)u\pi}{2N} \cdot \cos \frac{(2y+1)v\pi}{2N}
 \end{aligned} \tag{3.10}$$

逆变换为:

$$\begin{aligned}
 f(x,y) &= \frac{1}{N} F(0,0) + \frac{\sqrt{2}}{N} \sum_{v=1}^{N-1} F(0,v) \cos \frac{(2y+1)v\pi}{2N} \\
 &\quad + \frac{\sqrt{2}}{N} \sum_{u=1}^{N-1} F(u,0) \cos \frac{(2x+1)u\pi}{2N} \\
 &\quad + \frac{2}{N} \sum_{u=1}^{N-1} \sum_{v=1}^{N-1} F(u,v) \cos \frac{(2x+1)u\pi}{2N} \cdot \cos \frac{(2y+1)v\pi}{2N}
 \end{aligned} \tag{3.11}$$

更为简洁的定义方法是采用矩阵式定义。根据以上公式定义可知, 离散余弦变换的系数矩阵可以写成如下:

$$C = \sqrt{\frac{2}{N}} \begin{bmatrix} \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} & \cdots & \sqrt{\frac{1}{2}} \\ \cos \frac{1}{2N} \pi & \cos \frac{3}{2N} \pi & \cdots & \cos \frac{2N-1}{2N} \pi \\ \vdots & \vdots & \ddots & \vdots \\ \cos \frac{N-1}{2N} \pi & \cos \frac{3(N-1)}{2N} \pi & \cdots & \cos \frac{(2N-1)(N-1)}{2N} \pi \end{bmatrix} \tag{3.12}$$

3.3 基于离散余弦变换的图像数据的稀疏再表达方法

式(3.10) 和(3.12)中变换后的系数和系数矩阵是对原始信号(数据)的一种再表达。根据(3.11), 如果不计计算的精度损失, 这种再表达是完全可逆的。DCT 有一个很重要的性质, 即能量集中特性, 即大多数自然信号(声音、图像)的能量都集中在离散余弦变换后的低频部分, 因而 DCT 在(声音、图像)数据压缩中得到了广泛的使用。数据压缩的基本思路是将数据再表达后的数据进行量化, 然后进行熵编码, 这也就是静态图像压缩标准 JPEG(Joint Photographic Experts Group)所使用的方法。

JPEG 专家组开发了两种基本的压缩算法，一种是采用以离散余弦变换 (DCT) 为基础的有损压缩算法，另一种是采用以预测技术为基础的无损压缩算法。使用有损压缩算法时，在压缩比为 25:1 的情况下，压缩后还原得到的图像与原始图像相比较，非图像专家难于找出它们之间的区别，因此得到了广泛的应用。下图是基于二维离散余弦变换的 JPEG 流程示意图。

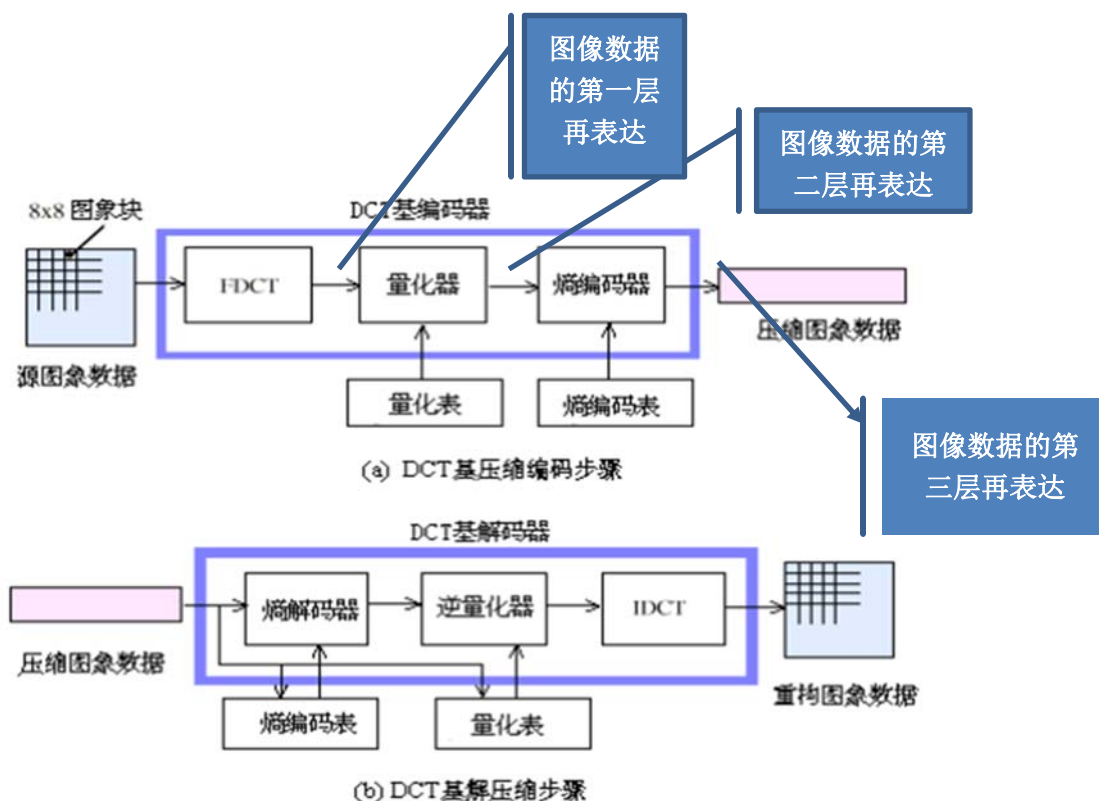


图 3.2 基于 DCT 编码和解码过程

其中的核心技术是二维离散余弦变换。二维离散余弦变换和量化运算的主要过程如下：

- (1) 将图像自上到下、自左至右分成 8×8 的图像块

$$\{f(x, y) : x, y = 0, 1, \dots, 63\};$$

- (2) 对每一个 8×8 的图像块做二维快速余弦变换，变换后的系数矩阵为

$$\{F(u, v) : u, v = 0, 1, \dots, 63\}; \text{ (第一层数据再表达)}$$

- (3) 给定量化的矩阵 $Q = (q_{uv}) \in Z^{64 \times 64}$ ，其中的元素为正整数，做量化运算：

$$\bar{F}(u, v) = \left\lfloor \frac{F(u, v)}{q(u, v)} \right\rfloor (u, v = 0, 1, \dots, 63) \text{ (第二层数据再表达——稀疏表达)}$$

注：(1) 如果不计精度损失，二维离散余弦变换和逆变换是无损可逆的，熵编码也是一个可逆变换。

(2) 量化过程 JPEG 唯一有损的过程是量化过程，量化矩阵一般采用专家建议的矩阵，量化的结果是使得一些变换后的系数（原始图像数据的再表达）变成了 0，这样后面的熵编码才能够提高编码效率，这也就使得逆量化器不能完全恢复原始信号。

量化后的系数 $\{\bar{F}(u,v):u,v=0,1,\dots,63\}$ 可以看成是对原始图像数据在余弦基下的一种再表达：

$$\begin{aligned} f(x,y) \approx & \frac{1}{N} \bar{F}(0,0) + \frac{\sqrt{2}}{N} \sum_{v=1}^{N-1} \bar{F}(0,v) \cos \frac{(2y+1)v\pi}{2N} \\ & + \frac{\sqrt{2}}{N} \sum_{u=1}^{N-1} \bar{F}(u,0) \cos \frac{(2x+1)u\pi}{2N} \\ & + \frac{2}{N} \sum_{u=1}^{N-1} \sum_{v=1}^{N-1} \bar{F}(u,v) \cos \frac{(2x+1)u\pi}{2N} \cdot \cos \frac{(2y+1)v\pi}{2N} \end{aligned} \quad (3.13)$$

由于量化后的系数 $\{\bar{F}(u,v):u,v=0,1,\dots,63\}$ 有很多零，所以变换系数的量化实质上是在二维 DCT 变换的数据的稀疏表示，也就是在该分析字典下的数据稀疏再表达，这实际上是一种模型驱动的数据的稀疏再表达方法。也就是说，JPEG 的核心技术是对图像数据的一种基于分析字典 DCT 的稀疏再表达方法。

(3) 熵编码是第三次数据再表达。

3.4 函数的多尺度逼近的数据再表达

离散的小波框架，其信息仍然是冗余的，从函数（信号）的再表达角度考虑，我们希望再表达后的数据尽量少的冗余。那么自然希望得到一组正交基，即正交小波基。在离散框架的基础上，取 $a_0=2, \Delta\tau=1$ ，则

$$\psi_{m,n}(t) = 2^{-m/2} \psi(2^{-m}t - n), \quad m, n \in Z \quad (3.14)$$

函数序列 $\{\psi_{m,n}(t)\}_{m,n \in Z}$ 能否构成一个再表达的正交基，如何构造母小波 $\psi(t)$ ？主要工具就是多分辨率分析。

给定函数（模拟信号） $f(t) \in L^2(\mathbb{R})$ ，可以用一串不同尺度 j 的函数序列 $\{f^{(j)}(t)\}$ 来逐步逼近，这种方法称为函数的多尺度逼近（或者称为函数多尺度再表达），这是模型驱动的数据再表达的一类重要的方法。例如，给定等距采样间隔 Δ 作为采样的基本单位，对应 j 尺度的采样间隔为 $\Delta\tau^{(j)} = \Delta\tau/2^j, j \in Z$ 。给定尺度 j ，根据采样间隔 $\Delta\tau^{(j)}$ 给出的分划节点为 $\{t_k^{(j)}\}$ ，对应的采样值为 $\{f(t_k^{(j)})\}$ ，基函数为 $\{\varphi_{j,k}(t)\}$ 。特别地，在多分辨率再表达中，我们选择的基函数为同一个函

数经过平移和尺度缩放后生成的函数序列, 如 $\varphi_{j,k}(t) = \varphi(2^j t - k)$ 。于是, 得到 $f(t)$

在 j 尺度下的近似函数 $f^{(j)}(t)$:

$$f^{(j)}(t) = \sum_{k \in \mathbb{Z}} c_{j,k} \varphi_{j,k}(t) \quad (3.15)$$

且

$$\lim_{j \rightarrow \infty} f^{(j)}(t) = \lim_{j \rightarrow \infty} \sum_{k \in \mathbb{Z}} c_{j,k} \varphi_{j,k}(t) = f(t) \quad (3.16)$$

由 (3.1) 和 (3.2), $\{c_{j,k}\}_{k \in \mathbb{Z}}$ 可以看成原始函数 $f(t)$ 是在 j 尺度下的在基 $\{\varphi_{j,k}(t)\}$ 下的近似再表达。

3.4.1 内插型逼近表示

若 $\{\varphi_{j,k}(t)\}$ 是内插型基函数, 即 $\varphi_{j,k}(t_m^j) = \delta_{k,m}$, 如 Shannon 采样定理中基函数、样条插值函数等, 则式 (3.1) 中的展开系数 $c_{j,k}$ 刚好是样本值 $f(t_k^{(j)})$ 。

3.4.2 非内插型逼近表示

如果基函数 $\{\varphi_{j,k}(t)\}$ 不满足 $\varphi_{j,k}(t_m^j) = \delta_{k,m}$, 则 $c_{j,k}$ 需要重新计算。此时, $c_{j,k}$ 就不能简单采用样本值 $f(t_k^{(j)})$ 。用到的基可以是正交基, 也可以是 Riesz 基, 并且可以在多尺度下逼近函数, 即多尺度下对原始信号进行逐层再表达。

3.4.3 函数的多尺度逼近表示

由于不同的基 $\{\varphi_{j,k}(t)\}$ 对应于不同的近似函数 $f^{(j)}(t)$, 即对应于不同再表达数据 $\{c_{j,k}\}_{k \in \mathbb{Z}}$ 。反过来, 给定基函数 $\{\varphi_{j,k}(t)\}$ 和尺度 j , 取不同的组合系数 $\{c_{j,k}\}_{k \in \mathbb{Z}}$, 由 (3.1) 式可以生成不同的函数 $f^{(j)}(t)$, 这些函数都属于 $L^2(\mathbb{R})$, 其全体记为:

$$V_j = \left\{ f^{(j)}(t) : f^{(j)}(t) = \sum_{k \in \mathbb{Z}} c_{j,k} \varphi_{j,k}(t), f^{(j)}(t) \in L^2(\mathbb{R}) \right\} \quad (3.17)$$

显然, V_j 是一个函数线性空间, 且 $V_j \subset L^2(\mathbb{R})$ 。再变动尺度 j , $\lim_{j \rightarrow \infty} f^{(j)}(t) = f(t)$

从函数子空间的角度可以描述为:

$$\cdots \subset V_j \subset V_{j+1} \subset \cdots \subseteq L^2(\mathbb{R})$$

所以, $\{V_j\}_{j \in \mathbb{Z}}$ 是一个嵌套的子空间逼近序列。对于给定的 $f(t) \in L^2(\mathbb{R})$, 在每个子空间 V_j 中取定一个关于 $f(t)$ 的近似函数 $f^{(j)}(t) \in L^2(\mathbb{R})$, 得到的近似函数序

列 $\{f^{(j)}(t)\}$ 就是逼近 $f(t)$ 的。

3.4.4 多尺度逼近表达的基本条件与正交逼近

多尺度逼近中的基函数 $\{\varphi_{j,k}(t)\}$ 和组合系数 $\{c_{j,k}\}_{k \in \mathbb{Z}}$ 按照 (3.1) 式构造的近似函数 $f^{(j)}(t) \in L^2(\mathbb{R})$ ，这种构造需要满足的基本条件是：

$$(1) \quad \{c_{j,k}\} \in l^2, \text{ 即 } \sum_{k \in \mathbb{Z}} |c_{j,k}|^2 < \infty$$

$$(2) \quad \text{基函数 } \{\varphi_{j,k}(t)\} \text{ 是 } L^2(\mathbb{R}) \text{ 空间的 Riesz 基, 即存在常数 } A, B \text{ 满足条件}$$

$$A \sum_{k \in \mathbb{Z}} |c_{j,k}|^2 \leq \left\| \sum_{k \in \mathbb{Z}} c_{j,k} \varphi_{j,k}(t) \right\|^2 \leq B \sum_{k \in \mathbb{Z}} |c_{j,k}|^2, \quad \{c_{j,k}\} \in l^2 \quad (3.4)$$

当然，基函数 $\{\varphi_{j,k}(t)\}$ 是 Riesz 基，可以不是平移正交的，也可以是平移正交的。若 $\{\varphi_{j,k}(t)\}$ 是标准正交基时，即

$$\langle \varphi_{j,k}(t), \varphi_{j,m}(t) \rangle = \delta_{km} = \begin{cases} 1, k = m \\ 0, k \neq m \end{cases} \quad (3.18)$$

则给近似函数 $f^{(j)}(t)$ 的表达带来方便：

$$f^{(j)}(t) = \sum_{k \in \mathbb{Z}} c_{j,k} \varphi_{j,k}(t), \quad f^{(j)}(t) \in V_j,$$

$$c_{j,k} = \langle f^{(j)}(t), \varphi_{j,k}(t) \rangle = \langle f(t), \varphi_{j,k}(t) \rangle,$$

$$\|f^{(j)}(t)\|^2 = \langle f^{(j)}(t), f^{(j)}(t) \rangle = \sum_{k \in \mathbb{Z}} |c_{j,k}|^2, \quad (3.19)$$

$$\langle f(t), \varphi_{j,k}(t) \rangle = \langle f^{(j+1)}(t), \varphi_{j,k}(t) \rangle = \langle f^{(j)}(t), \varphi_{j,k}(t) \rangle = c_{j,k}. \quad (3.20)$$

3.3.5 多分辨率分析与图像数据的逐层再表达

1. 多分辨率分析

多尺度逼近是指一串嵌套式子空间序列 $\{V_j\}_{j \in \mathbb{Z}}$ 满足下列条件：

$$(1) \quad \cdots \subset V_j \subset V_{j+1} \subset \cdots \subseteq L^2(\mathbb{R}), \bigcap_{j \in \mathbb{Z}} V_j = \{0\}, \bigcup_{j \in \mathbb{Z}} V_j = L^2(\mathbb{R}); \quad (3.21)$$

$$(2) \quad V_j = \text{span}\{\varphi_{j,k}(t) : \varphi_{j,k}(t) = \varphi(2^j t - k), k \in \mathbb{Z}\}; \quad (3.22)$$

$$(3) \quad \{\varphi_{j,k}(t)\} \text{ 是 } L^2(\mathbb{R}) \text{ 空间的 Riesz 基}.$$

设 $\varphi(t)$ 生成了一个多分辨率分析 $\{V_j\}_{j \in \mathbb{Z}}$ ，尺度函数 $\varphi(t)$ 导出 V_0 ， $\varphi(t) \in V_0$ 。由

于 $V_0 \subset V_1$, 所以 $\varphi(t) \in V_1$, 因此, $\varphi(t)$ 能用 V_1 的基 $\{\varphi_{1,k}(t): \varphi_{1,k}(t) = \sqrt{2}\varphi(2t-k), k \in Z\}$ 重新表达, 即存在序列 $\{h_n\} \in l^2$ 使得

$\varphi(t) = \sum_{n \in Z} h_n \varphi(2t-n)$ 。所以, 多分辨率分析是指一串嵌套式子空间序列 $\{V_j\}_{j \in Z}$ 满足下列条件:

$$(1) \quad \cdots \subset V_j \subset V_{j+1} \subset \cdots \subseteq L^2(\mathbb{R}), \bigcap_{j \in Z} V_j = \{0\}, \bigcup_{j \in Z} V_j = L^2(\mathbb{R}); \quad (3.23)$$

$$(2) \quad V_j = \text{span}\{\varphi_{j,k}(t): \varphi_{j,k}(t) = 2^{j/2}\varphi(2^j t - k), k \in Z\}; \quad (3.24)$$

$$(3) \quad \varphi(t) = \sum_{n \in Z} h_n \varphi(2t-n), \{h_n\} \in l^2; \quad (3.25)$$

(4) $\{\varphi(t-k)\}$ 是 $L^2(\mathbb{R})$ 空间的 **Riesz** 基。

多分辨率分析是在多尺度逼近的基础上增加了少量条件。取

$$\{\varphi_{j,k}(t): \varphi_{j,k}(t) = 2^{j/2}\varphi(2^j t - k), k \in Z\} \quad (3.26)$$

是为了使得 $\{\varphi_{j,k}(t): k \in Z\}$ 成为标准基 ($\|\varphi_{j,k}(t)\|^2 = 1$)。

由函数 $\varphi(t)$ 生成了 $\{V_j\}_{j \in Z}$, 从而生成了多分辨率分析, (3.10) 称为双尺度方程, $\varphi(t)$ 称为尺度函数。双尺度方程是为了表述子空间序列 $\{V_j\}_{j \in Z}$ 之间的关系, 有利于实现快速算法。

2. 函数的逐层再表达

对于 $\varphi(t)$ 生成的多分辨率分析 $\{V_j\}_{j \in Z}$, 函数 $f(t)$ 从子空间 V_j 中的表达到 V_{j+1} 中的表达, 对 $f(t)$ 的逼近更好, 即相比于在 V_j 中的表达, 在 V_{j+1} 中的表达更增加了一些细节。下面就考虑这些增加的细节部分, 这些细节部分可以用“小波”来表达。由于 $V_0 \subset V_1$, 考虑 V_0 关于 V_1 的补空间 W_0 :

$$V_1 = V_0 \cup W_0, \quad V_0 \cap W_0 = \{0\} \quad (3.27)$$

即 V_1 是 V_0 和 W_0 的直和: $V_1 = V_0 \oplus W_0$ 。为了构造生成 W_0 的“小波” $\psi(t)$, 使得类似于 $\varphi(t)$ 生成 V_0 , 由于 $\psi(t) \in W_0 \subset V_1$, $\psi(t)$ 也能用 V_1 的基

$$\{\varphi_{1,k}(t): \varphi_{1,k}(t) = \sqrt{2}\varphi(2t-k), k \in Z\}$$

重新表达, 即存在序列 $\{g_n\} \in l^2$ 使得:

$$\psi(t) = \sum_{n \in \mathbb{Z}} g_n \varphi(2t - n) \quad (3.28)$$

(3.28)式称为 $\psi(t)$ 与 $\varphi(t)$ 的两尺度关系。以此类推,得到空间序列 $\{W_j\}_{j \in \mathbb{Z}}$, 并且 $W_j = \text{span}\{\psi_{j,k}(t) : \psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k), k \in \mathbb{Z}\}$, 称为尺度为 j 的小波空间。由此可以看出, 给出满足一定条件的两个序列 $\{h_n\}, \{g_n\} \in l^2$, 由双尺度方程

$$\begin{cases} \varphi(t) = \sum_{n \in \mathbb{Z}} h_n \varphi(2t - n) \\ \psi(t) = \sum_{n \in \mathbb{Z}} g_n \varphi(2t - n) \end{cases} \quad (3.29)$$

就完全确定了尺度函数与小波函数 $\varphi(t)$ 和 $\psi(t)$, 并且 $\varphi(t)$ 和 $\psi(t)$ 的分解关系为:

$$\varphi(2t - l) = \sum_{n \in \mathbb{Z}} \{a_{l-2n} \varphi(t - n) + b_{l-2n} \psi(t - n)\}, l = 0, \pm 1, \pm 2, \dots \quad (3.30)$$

其中

$$a_n := \frac{1}{2} g_{-n}, \quad b_n := \frac{1}{2} h_{-n} \quad (3.31)$$

下面我们就给出基于多分辨率分析的一般信号的逐层再表达和重建算法, 然后给出数字图像数据的逐层再表达和重建算法。

显然 $L^2(\mathbb{R}) = \dots \oplus V_{-1} \oplus V_0 \oplus V_1 \oplus \dots$, 所以, $\forall f(t) \in L^2(\mathbb{R})$, 都有唯一分解

$$f(t) = \sum_{k=-\infty}^{\infty} g_k(t) \quad (3.32)$$

其中 $g_k(t) \in W_k (k \in \mathbb{Z})$ 。令

$$f_{k+1}(t) = \sum_{n=-\infty}^k g_n(t) \quad (3.33)$$

则有 $f_k(t) \in V_k$ 且

$$f_{k+1}(t) = g_k(t) + f_k(t) \quad (3.34)$$

其中 $f_k(t) \in V_k$, $g_k(t) \in W_k$ 。所以可以用它们的基表达为:

$$f_k(t) = \sum_{n=-\infty}^k c_{k,n} \varphi(2^k t - n) \quad (3.35)$$

$$g_k(t) = \sum_{n=-\infty}^k d_{k,n} \psi(2^k t - n) \quad (3.36)$$

由 (3.34)、(3.35) 和 (3.36) 可得：

$$f_{k+1}(t) = \sum_{n=-\infty}^k c_{k,n} \varphi(2^k t - n) + \sum_{n=-\infty}^k d_{k,n} \psi(2^k t - n) \quad (3.37)$$

由于 $f_{k+1}(t)$ 是函数 $f(t)$ 在 V_{k+1} 子空间的近似，所以由 (3.37) 式，序列 $\{c_{k,n}, d_{k,n}\}_{n \in \mathbb{Z}}$ 是函数 $f(t)$ 的一种近似再表达。当尺度 k 发生变化时， $f(t)$ 的这种近似再表达也随着变化。

基于正交的数据逐层再表达算法就是要在 $\{\varphi_{k,n} : n \in \mathbb{Z}\}, \{\psi_{k,n} : n \in \mathbb{Z}\}$ 是标准正交基条件下，已知双尺度方程 (3.29) 中的系数 $\{h_n\}_{n \in \mathbb{Z}}$ 和 $\{g_n\}_{n \in \mathbb{Z}}$ ，由函数 $f(t)$ 在前一层（尺度 $k+1$ ）上的近似再表达的主要（低频）成分 $\{c_{k+1,n}\}_{n \in \mathbb{Z}}$ ，计算出在当前层（尺度 k ）上的近似再表达的主要（低频）成分 $\{c_{k,n}\}_{n \in \mathbb{Z}}$ 和细节（高频）成分 $\{d_{k,n}\}_{n \in \mathbb{Z}}$ ，其计算公式为：

$$\begin{aligned} c_{k,n} &= 2^{-1/2} \sum_l c_{k+1,l} h_{l-2n} \\ d_{k,n} &= 2^{-1/2} \sum_l c_{k+1,l} g_{l-2n} \end{aligned} \quad (3.38)$$

$\{h_n\}_{n \in \mathbb{Z}}$ 可以看成是低通滤波器，而 $\{g_n\}_{n \in \mathbb{Z}}$ 以看成是高通滤波器。逐层再表达过程为：

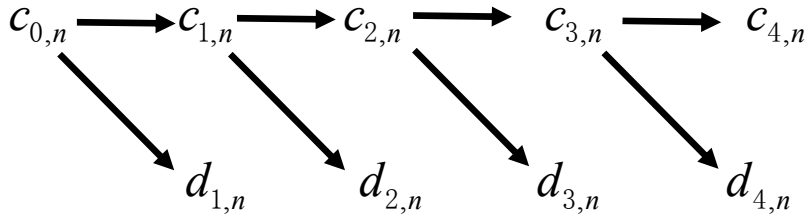


图 3.3 基于高通和低通滤波器的数据逐层再表达过程

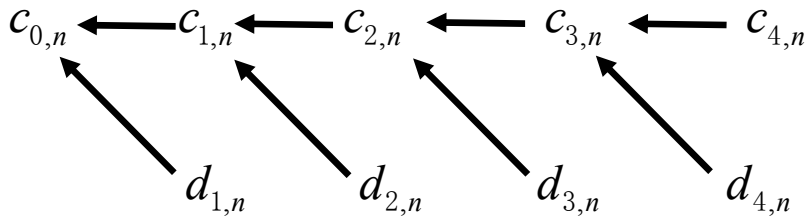


图 3.4 数据的逐层再表达的逐层重构流程

在实际应用中, 假定函数 $f(t)$ 的近似水平为 $f_N(t)$, 即 $f(t) \approx f_N(t)$, 那么就可以根据上述过程逐层进行对应数据的再表达。下面给出重构公式。

重构算法是再表达算法的逆过程。此时, 已知函数 $f(t)$ 的再表达数据 $\{c_{k,n}\}_{n \in \mathbb{Z}}$, $\{d_{k,n}\}_{n \in \mathbb{Z}}$, $0 \leq k \leq M-1$, 希望利用这些数据快速准确地重构出后一层的再表达数据 $\{c_{M,n}\}_{n \in \mathbb{Z}}$ 。重构公式为:

$$c_{k+1,n} = 2^{-1/2} \sum_l c_{k,(n-l)/2} h_l + 2^{-1/2} \sum_l d_{k,(n-l)/2} g_l \quad (n \in \mathbb{Z}) \quad (3.39)$$

3.5 图像数据的逐层稀疏再表达与图像数据压缩

将上述数据的逐层再表达和逐层重构方法推广到二维情形, 逐层再表达公式为:

$$\left\{ \begin{aligned} C_{j+1}(m,n) &= \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} h^*(k-2m) h^*(l-2n) C_j(k,l) \\ D_{j+1}^1(m,n) &= \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} h^*(k-2m) g^*(l-2n) C_j(k,l) \\ D_{j+1}^2(m,n) &= \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} g^*(k-2m) h^*(l-2n) C_j(k,l) \\ D_{j+1}^3(m,n) &= \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} g^*(k-2m) g^*(l-2n) C_j(k,l) \end{aligned} \right. \quad (3.40)$$

其中, C_{j+1} 是 C_j 的低频近似分量部分, D_{j+1}^1 是水平方向上的高频部分, D_{j+1}^2 是垂直方向上的高频部分, D_{j+1}^3 是对角线方向上的高频部分。

相应地, 逐层重构公式为:

$$\begin{aligned} C_j(k,l) &= \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} h(k-2m) h(l-2n) C_{j+1}(m,n) \\ &+ \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} h(k-2m) g(l-2n) D_{j+1}^1(m,n) \\ &+ \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} g(k-2m) h(l-2n) D_{j+1}^2(m,n) \\ &+ \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} g(k-2m) g(l-2n) D_{j+1}^3(m,n) \end{aligned} \quad (3.41)$$

图像数据的二维离散小波变换实质上是对图像的一种再表达, 而逆变换则是

其重构。将上述二维数据的逐层再表达和逐层重构方法用到图像数据上，如下图所示：

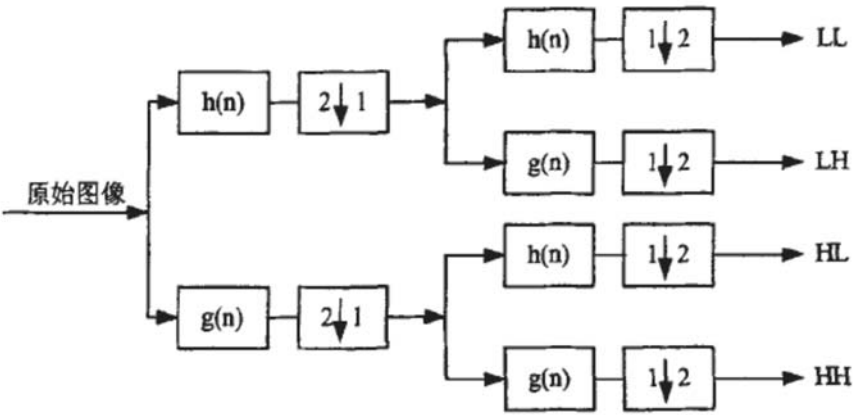


图 3.5 图像数据逐层再表达中的一层算法示例

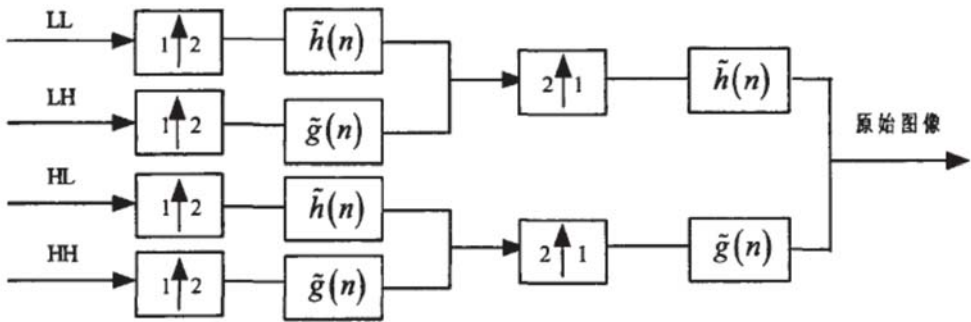


图 3.6 图像数据逐层重建中的一层算法示例

从上面的图像的逐层表达和重构过程可以看出，图像在经过多层再表达之后，再表达的数据实质上是不同分辨率及不同方向的多个子带分量。下图是图像的三层再表达示意图。

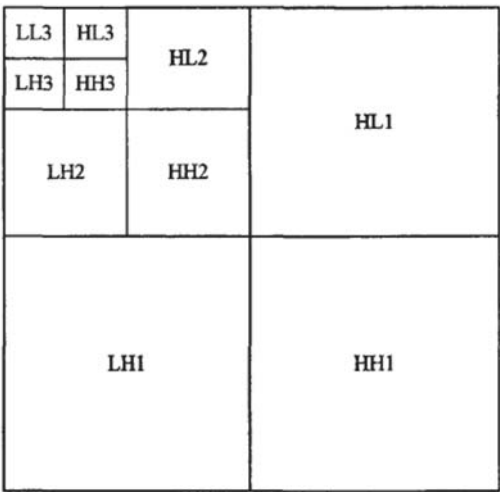


图 3.7 逐层小波分解

上面图像小波分解与重构当中的 LL 表示的是图像的低频分量，它描绘的是图

像的轮廓信息；而 LH 、 HL 和 HH 表示的是图像在水平、垂直以及对角线方向上的分量，它们描绘的是图像在这三个方向上的细节信息。

下图是原始图像和一层再表达的数据。



原始图像

一层再表达的数据

图 3.8 始图像和一层再表达的数据

可以看到，一层再表达后，将一幅图像通过二维小波变换分解成了一系列方向和空间局部变化的子带，得到一系列不同分辨率的子图像。图像的能量主要集中在低频子带 LL 中，其它高频子带绝大多数系数都接近 0。这就为图像数据的稀疏再表达提供了可能，而图像数据的稀疏再表达是图像数据压缩的关键。

图像的小波压缩过程首先是对原始图像进行二维小波变换，得到小波变换系数。由于小波变换能将原始图像的能量集中到少部分小波系数上，且分解后的小波系数在三个方向的细节分量有高度的局部相关性，为进一步量化和编码提供了条件，因此基于这种稀疏再表达的方法可以获得较高的压缩比。图 3.8 是基于小波变换的逐层数据再表达的图像压缩流程：

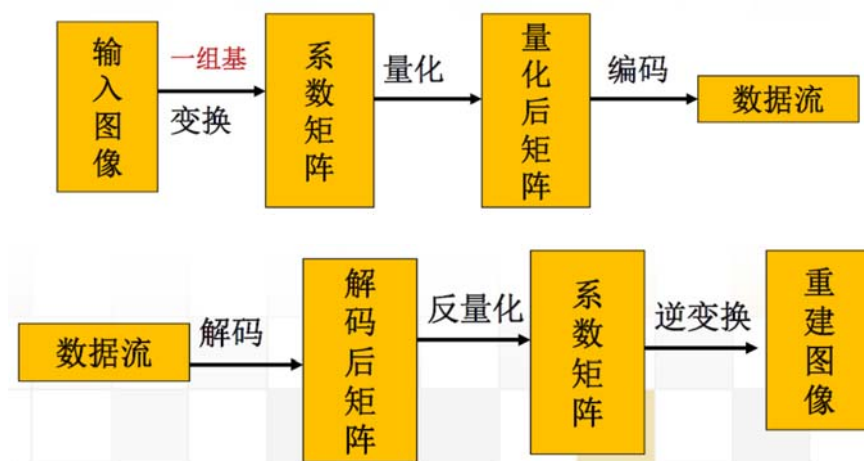


图 3.9 基于小波变换的逐层数据再表达的图像压缩和重建流程

图中数据再表达是两层再表达示意图，第一层表达为低频成分 $LL1$ 和三个方向的高频成分 $HL1$ 、 $LH1$ 、 $HH1$ 。第二层表达时，把低频成分 $LL1$ 再表达为

低频成分 LL2 和三个方向的高频成分 HL2、LH2、HH2。在第二层再表达时，可以选择对第一层表达的三个方向的高频成分是否也进行再表达（用虚线表示可以选择是否进行分解），其它层再表达时以此类推。量化步骤类似于 JPEG，对不同频带可以选择不同的量化矩阵或者量化方法，目标是使得图像的稀疏再表达数据，重构后图像损失最小（信噪比最大）。解压缩是压缩的逆过程，在此不再赘述。

另外，利用图像的多尺度逐层再表达后的数据，可以进一步检测图像边缘、确定边缘奇异性等特征。