Lecture 3

LSI systems and convolution in N-D

3.1 Learning Objectives

- Understand the extension of linear shift invariant systems to the multi-dimensional case.
- Appreciate the pervasive utility of multidimensional convolutions (and therefore LSI systems) in various applications, including imaging and therapy.

3.2 Linear Systems

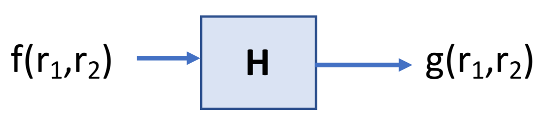

Figure 3.1: Example multi-dimensional system H, which takes in an input signal f and produces an output g.

Similar to the 1-D case, a subset of N-D systems of particular relevance to signal processing are linear systems, where if $f_1(r_1, r_2) \longrightarrow g_1(r_1, r_2)$ and $f_2(r_1, r_2) \longrightarrow g_2(r_1, r_2)$, then:

$$a_1 \cdot f_1(r_1, r_2) + a_2 \cdot f_2(r_1, r_2) \longrightarrow a_1 \cdot g_1(r_1, r_2) + a_2 \cdot g_2(r_1, r_2)$$

for any input signals $f_1(r_1, r_2)$ and $f_2(r_1, r_2)$ and scalars a_1 and a_2.

3.3 Shift-Invariant Systems

Another subset of systems we care about are shift-invariant systems, where if $f_1 \longrightarrow g_1$ and $f_2(r_1, r_2) = f_1(r_1 - s_1, r_2 - s_2)$ (ie: f_2 is a shifted version of f_1), then:

$$f_2(r_1, r_2) \longrightarrow g_2(r_1, r_2) = g_1(r_1 - s_1, r_2 - s_2)$$

for any input signals f_1 and shifts s_1 and s_2.

In other words, for N-D shift-invariant systems, if we shift the input in (our N-dimensional) space, the output shifts in space accordingly.

3.4 LSI Systems

Linear shift-invariant systems are systems that satisfy both of the properties described above: linearity and shift-invariance.

Question: Which of the following N-D systems are linear? Which are shift-invariant?

A. $g(r_1, r_2) = f(r_1, r_2)$

B. $g(r_1, r_2) = |f(r_1, r_2)|$

C. $g(r_1, r_2) = \max(0, f(r_1, r_2))$, aka rectified linear unit (ReLU) in machine learning lingo (for real-valued signals)

D. $g(r_1, r_2) = f(r_1, r_2) + 1$

E. $g(r_1, r_2) = f(0, 0)$

F. $g(r_1, r_2) = \int_{s_1=-1}^{+1} \int_{s_2=-1}^{+1} f(r_1 - s_1, r_2 - s_2) ds_1 ds_2$

G. $g(r_1, r_2) = \int_{s_1=-\infty}^{+\infty} \int_{s_2=-\infty}^{+\infty} f(s_1, s_2) e^{-i2\pi(s_1 r_1 + s_2 r_2)} ds_1 ds_2$

3.5 Convolution

The N-D convolution is an operation between two signals f_1 and f_2 defined as:

$$[f_1 * f_2](r_1, r_2) = \int_{s_1=-\infty}^{+\infty} \int_{s_2=-\infty}^{+\infty} f_1(r_1 - s_1, r_2 - s_2) f_2(s_1, s_2) ds_1 ds_2 \tag{3.1}$$

Similarly to the 1-D case, in N-D systems that are LSI, we can simply express the output of the system as a convolution between the input and the impulse response. This can be written as:

$$f(r_1, r_2) \longrightarrow [f * h](r_1, r_2) \tag{3.2}$$

3.6 Examples of N-D LSI Systems and Convolution

3.6.1 Separable Systems

Certain N-dimensional signals can be decomposed as a product of 1D systems. For instance, a 2D signal $h(x, y)$ that can be decomposed as $h(x, y) = h_1(x) \cdot h_2(y)$ is termed separable. Importantly, convolution with a separable signal can be performed separately (sequentially) along each dimension. This is illustrated in Figure 3.2.

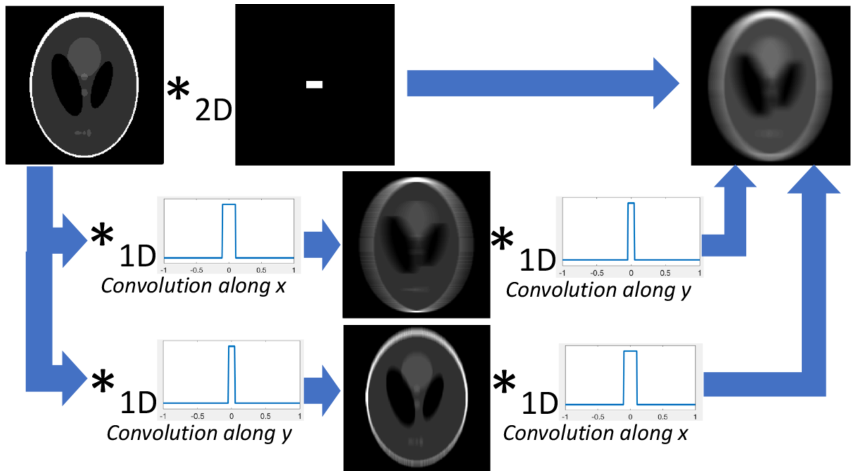

Figure 3.2: 2D convolution with a separable signal $h(x, y) = h_1(x) \cdot h_2(y)$ can be done as two sequential 1D convolutions, with $h_1(x)$ along x, and with $h_2(y)$ along y, respectively.

3.6.2 Convolution Layers in Convolutional Neural Nets

The multi-dimensional convolution operation is central to modern machine learning based image analysis algorithms, including classification of images (see Figure 3.3). In this case, the convolution filter is itself trained (as well as a multitude of other CNN parameters), ie: these parameters are derived from large, typically pre-labeled, datasets.

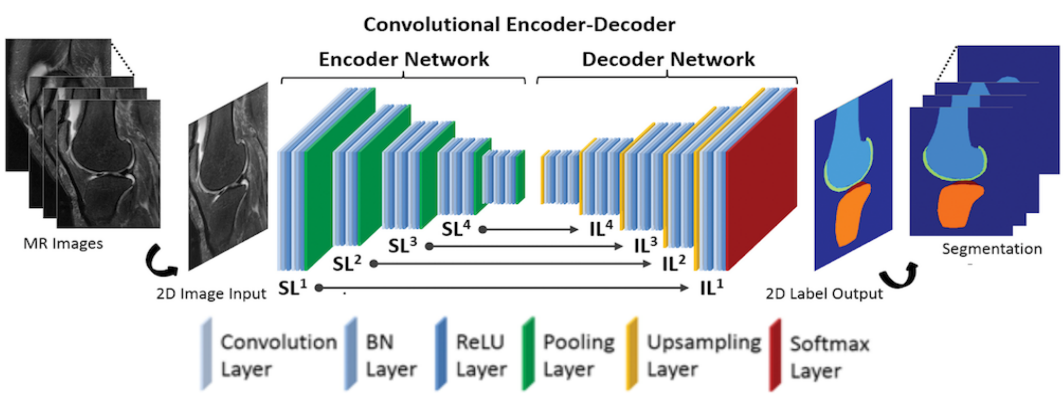

Figure 3.3: Example Convolutional Neural Network (CNN) used for segmentation of knee MR images. Image courtesy of Fang Liu, PhD (F. Liu et al, Deep Learning Approach for Evaluating Knee MR Images: Achieving High Diagnostic Performance for Cartilage Lesion Detection, Radiology 2018).

3.6.3 Convolution in Therapy Planning

Accurate determination of absorbed dose is central to radiotherapy applications. Under certain assumptions, the absorbed dose can be efficiently calculated using convolutions (see example in 3.4). A number of UW investigators have contributed to this area.

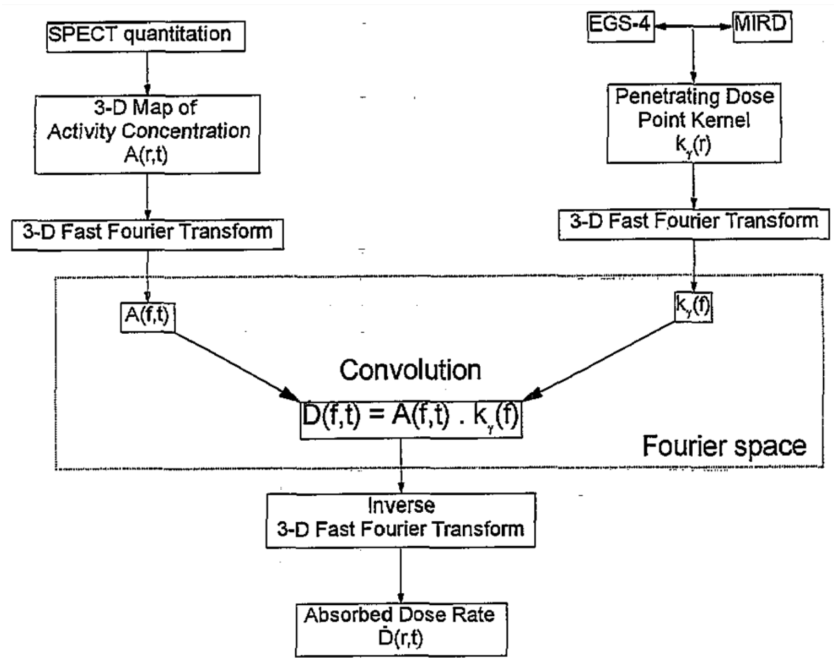

Figure 3.4: Dose calculation algorithm based on 3D convolution, which itself is calculated using 3D FFTs (algorithm described in HB Giap et al, Physics in Medicine and Biology 1995, 40:365-381). We will study FFTs later in this course.