Convolution

L06

Goal: To define convolution, see what it does, and find out how it can be done using the Fourier transform.

Def'n: The convolution between two functions $f(x)$ and $g(x)$ is an integral of the form

$$(f*g)(x) = \int_{-\infty}^{\infty} f(\tau)\, g(x-\tau)\, d\tau$$

For discrete signals f_n and g_n,

$$(f*g)_n = \sum_{k=0}^{N-1} f_k\, g_{n-k} \qquad n=0,\ldots,N-1$$

Theorem: (continuous-domain version)

Let $f(x)$ and $g(x)$ be functions, and let

$$F(\omega) = \mathcal{F}\{f(x)\}(\omega) \quad \text{and} \quad G(\omega) = \mathcal{F}\{g(x)\}(\omega)$$

be their Fourier transforms. Then

$$\mathcal{F}\{(f*g)(x)\}(\omega) = \mathcal{F}\{f(x)\}(\omega)\cdot\mathcal{F}\{g(x)\}(\omega)$$

$$= F(\omega)\, G(\omega)$$

Pf: $\mathcal{F}\{(f*g)(x)\}(\omega) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\tau)\, g(x-\tau)\, d\tau\, e^{-2\pi i \omega x}\, dx$

Change of variables: Let $y = x - \tau \Rightarrow x = y + \tau$

$dy = dx$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\tau)\, g(y)\, d\tau\, e^{-2\pi i w(y+\tau)}\, dy$$

$$= \int_{-\infty}^{\infty} f(\tau)\, e^{-2\pi i w \tau}\, d\tau \int_{-\infty}^{\infty} g(y)\, e^{-2\pi i w y}\, dy$$

$$= F(w)\; G(w)$$

$\blacksquare$

Note: One can just as easily prove the theorem

$$\mathcal{F}^{-1}\{(F*G)(w)\}(x) = f(x)\, g(x)$$

$\therefore$ convolution in one domain is equivalent to element-wise multiplication in the other domain.

$\mathcal{F}\{f*g\}(w) = F(w)\, G(w)$

conv. in spatial domain
$\Leftrightarrow$ mult. in freq. domain

$\mathcal{F}^{-1}\{F*G\}(x) = f(x)\, g(x)$

conv. in freq. domain
$\Leftrightarrow$ mult. in spatial domain.

Theorem: (discrete-domain version)

Let f_n and g_n, $n = 0, \ldots, N-1$, be two discrete functions (signals).

$$\mathcal{F}\{(f*g)_n\}_k = F_k\, G_k$$

TASK: Check out the Matlab script
L06_Convolution.m

<u>Question</u>: Why bother using the FT to compute convolution? Sure, it may be cool, but isn't it more work?

Consider the # of flops (floating-point operations) to compute $(f*g)$ using the 2 methods.

$$\begin{tikzcd}[column sep=huge, row sep=large]
f_n, g_n \arrow[r, "(f*g)_n = \sum_{k=0}^{N-1} f_k g_{n-k} \quad n=0,\dots,N-1", "\text{Direct comp.} \to O(N^2)"'] \arrow[d, "O(N\log N)"'] \arrow[r, bend right=60, "O(N\log N)"'] & f*g \\
F_k, G_k \arrow[r, "O(N)"'] & F_k \cdot G_k \arrow[u, "O(N\log N)"']
\end{tikzcd}$$

Thus, as N gets large, using the DFT is more efficient.
What about convolving $N \times N$ images, f_{mn} and g_{mn}, where $m = 0, \dots, N-1$ and $n = 0, \dots, N-1$?

$\to O(N^2 \log N)$ flops