Discrete convolution

And how to calculate it with the FFT

Herman Kamper

Discrete convolution

Continuous:

$$h(t) * x(t) = \int_{-\infty}^{\infty} h(\tau) . x(t-\tau) . d\tau$$

Definition:

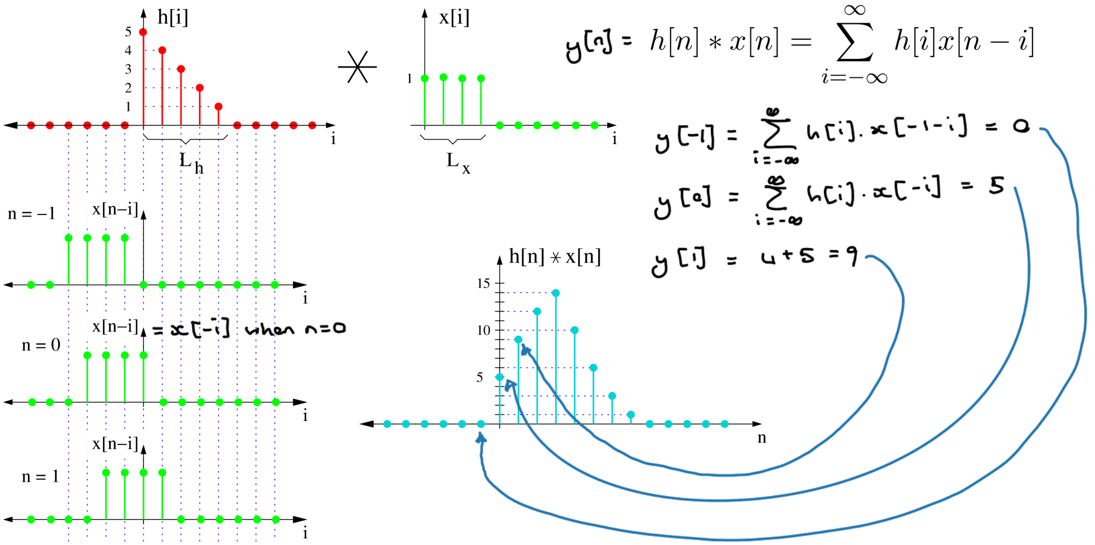
$$h[n] * x[n] = \sum_{i=-\infty}^{\infty} h[i]x[n-i]$$

Properties:

- Commutative: $x[n] * h[n] = h[n] * x[n]$
- Associative: $\Big[x[n] * h_1[n]\Big] * h_2[n] = x[n] * \Big[h_1[n] * h_2[n]\Big]$
- Distributive: $x[n] * \Big[h_1[n] + h_2[n]\Big] = x[n] * h_1[n] + x[n] * h_2[n]$

Discrete convolution example

$$y[n] = h[n] * x[n] = \sum_{i=-\infty}^{\infty} h[i]x[n-i]$$

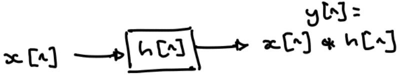


$$y[-1] = \sum_{i=-\infty}^{\infty} h[i] \cdot x[-1-i] = 0$$

$$y[0] = \sum_{i=-\infty}^{\infty} h[i] \cdot x[-i] = 5$$

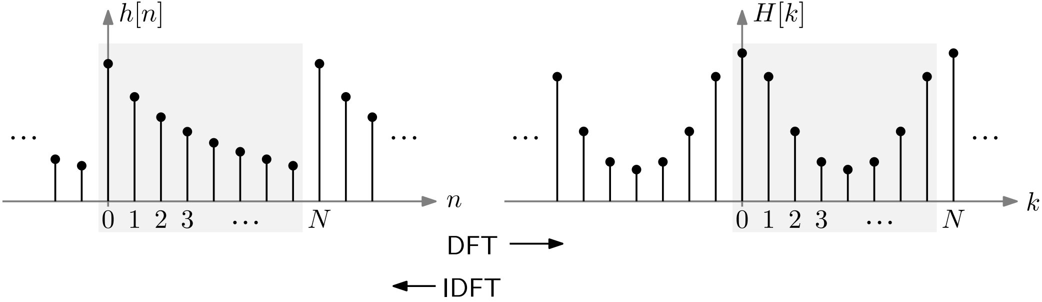
$$y[1] = 4 + 5 = 9$$

Why and how?

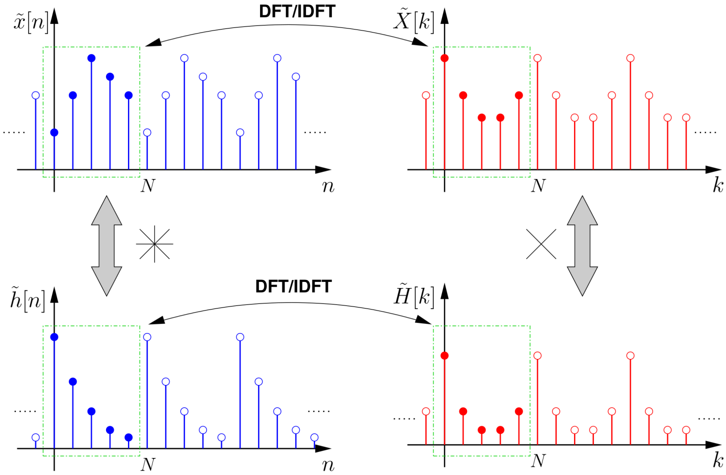


- Why this obsession with convolution?
- Direct computation of discrete convolution takes roughly (a ton of) N^2 multiplications
- Can we do this using the FFT? $h(t) * x(t) \Leftrightarrow H(f) \cdot X(f)$
  - Calculate $H[k]$ and $X[k]$
  - Multiply in frequency domain
  - Take N-point IDFT
- Would be more efficient: $H[k]$, $X[k]$, and IDFT each take $\frac{N}{2}\log_2 N$ mults
- But the following is unfortunately not actually true:

$$x[n] * h[n] \not\Leftrightarrow X[k] \cdot H[k]$$


h[n]
n
0 1 2 3 ⋯ N
H[k]
k
0 1 2 3 ⋯ N
DFT
IDFT

Discrete convolution and the DFT



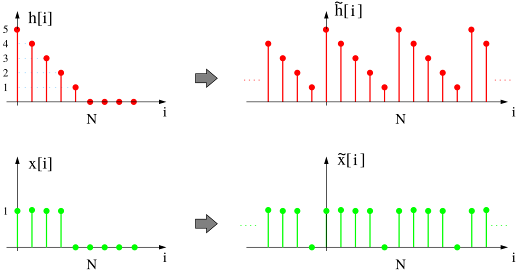
So what happens in time domain when we multiply DFTs?

Circular convolution:

$$H[k] \cdot X[k] \iff \quad h[n] \underset{N}{\circledast} x[n] = \sum_{i=0}^{N-1} h[i]\tilde{x}[n-i]$$

$$= \sum_{i=0}^{N-1} x[i]\tilde{h}[n-i]$$

where $\tilde{x}[n]$ and $\tilde{h}[n]$ are periodic extensions of $x[n]$ and $h[n]$:



Multiplication in the time-domain leads to circular convolution in the frequency domain:

$$\mathrm{DFT}\{x[n]y[n]\} = \frac{1}{N}\mathrm{DFT}\{x[n]\} \underset{N}{\circledast} \mathrm{DFT}\{y[n]\}$$

Multiplication in the frequency domain leads to circular convolution in the time-domain:

$$\mathrm{DFT}\{x[n] \underset{N}{\circledast} y[n]\} = \mathrm{DFT}\{x[n]\} \cdot \mathrm{DFT}\{y[n]\}$$

Proof: multiplying DFTs match circular convolution in time

You can go through the proof on your own.

Suppose that we have two finite-duration sequences of length N, $x_1[n]$ and $x_2[n]$. Their N-point DFTs are

$$X_1[k] = \sum_{n=0}^{N-1} x_1[n] e^{-j2\pi nk/N} \qquad \text{for } k = 0, 1, \ldots, N-1$$

$$X_2[k] = \sum_{n=0}^{N-1} x_2[n] e^{-j2\pi nk/N} \qquad \text{for } k = 0, 1, \ldots, N-1$$

If we multiply the two DFTs together, the result is a DFT, call it $X_3[k]$, of a sequence $x_3[n]$ of length N. What is the relationship between $x_3[n]$ and the sequences $x_1[n]$ and $x_2[n]$?

We have

$$X_3[k] = X_1[k] \cdot X_2[k] \qquad \text{for } k = 0, 1, \ldots, N-1$$

The IDFT of $X_3[k]$ is

$$\begin{aligned} x_3[m] &= \frac{1}{N} \sum_{k=0}^{N-1} X_3[k] e^{j2\pi km/N} \\ &= \frac{1}{N} \sum_{k=0}^{N-1} X_1[k] X_2[k] e^{j2\pi km/N} \end{aligned}$$

Suppose that we substitute for $X_1[k]$ and $X_2[k]$ using the DFTs given above:

$$\begin{aligned} x_3[m] &= \frac{1}{N} \sum_{k=0}^{N-1} \left[\sum_{n=0}^{N-1} x_1[n] e^{-j2\pi nk/N} \right] \left[\sum_{l=0}^{N-1} x_2[l] e^{-j2\pi kl/N} \right] e^{j2\pi km/N} \\ &= \frac{1}{N} \sum_{n=0}^{N-1} x_1[n] \sum_{l=0}^{N-1} x_2[l] \left[\sum_{k=0}^{N-1} e^{j2\pi k(m-n-l)/N} \right] \end{aligned}$$

The inner sum in the brackets has the form

$$\sum_{k=0}^{N-1} a^k = \begin{cases} N & \text{for } a = 1 \\ \dfrac{1-a^N}{1-a} & \text{for } a \neq 1 \end{cases}$$

where $a = e^{j2\pi(m-n-l)/N}$.

We observe that $a = 1$ when $m - n - l$ is a multiple of N. On the other hand, $a^N = 1$ for any value of $a \neq 0$. Consequently,

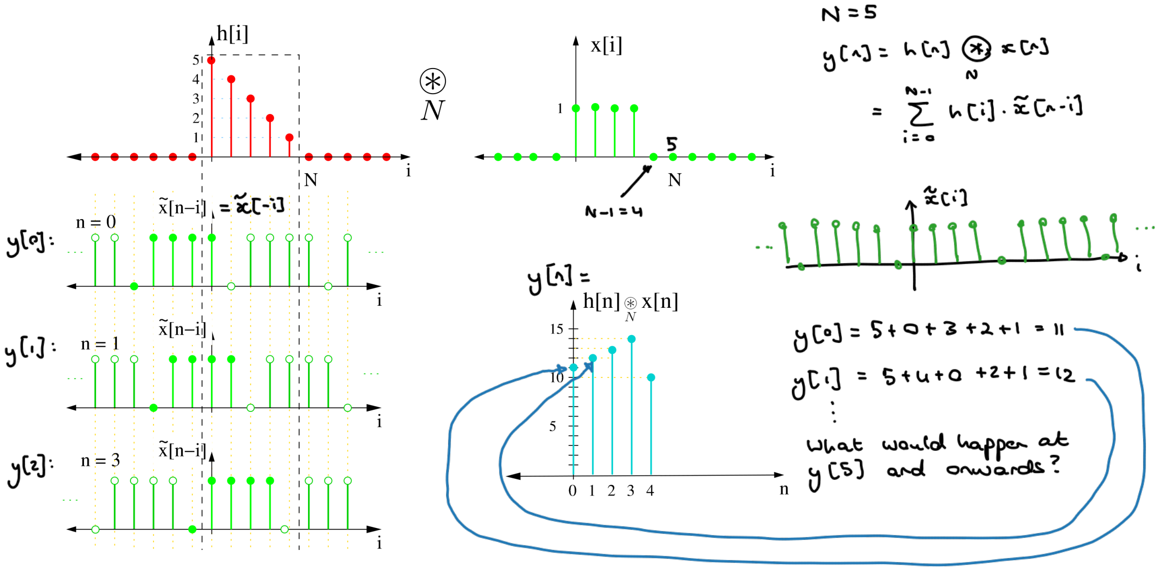
$$\sum_{k=0}^{N-1} a^k = \begin{cases} N & \text{for } l = m - n + pN, \quad p \text{ an integer} \\ 0 & \text{otherwise} \end{cases}$$

If we substitute this result into the expression above, we obtain the expression for $x_3[m]$:

$$x_3[m] = \sum_{n=0}^{N-1} x_1[n]\, \tilde{x}_2[m-n] \qquad \text{for } m = 0, 1, \ldots, N-1$$

This is the circular convolution.

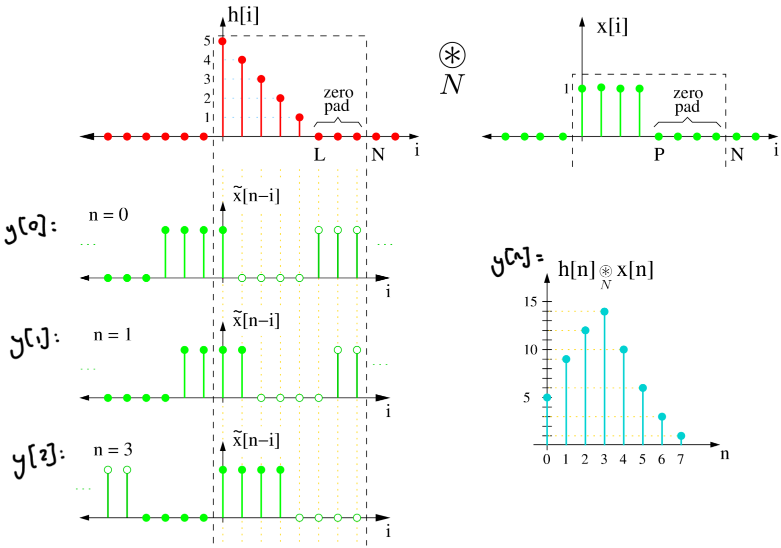
Circular convolution example



Zero padding: Discrete convolution via circular convolution

- Signal $x[n]$ has length L and signal $h[n]$ has length P
- Zero pad both to length N so that $N \geq L + P - 1$
- Calculate N-point DFT for both
- Multiply DFTs
- Calculate the inverse DFT: Result is the discrete convolution (not circular)

Zero padding example



- L = 5, P = 4
- Zero pad to
  $N \geq L + P - 1 = 8$
  Choose $N = 8$

(if not power of 2, rather go up to power of 2)

$y[0] = 5$

$y[1] = 5 + 4 = 9$

$y[2] =$

⋮

How does this compare to the figure on slide 3?

Discrete convolution example

$$x[n] = \{\ \underset{\uparrow}{1}\quad 2\quad 3\quad 1\ \}\qquad y[n] = \{\ \underset{\uparrow}{2}\quad 3\quad 1\quad 2\ \}$$

What is $x[n] * y[n]$?

$$x[n] * y[n] = \{\ 0\quad \underset{\uparrow}{2}\quad 7\quad 13\quad 15\quad 10\quad \cancel{10}\quad 7\quad 2\ \}$$

Circular convolution example

$$x[n] = \{ \underset{\uparrow}{1} \quad 2 \quad 3 \quad 1 \} \qquad y[n] = \{ \underset{\uparrow}{2} \quad 3 \quad 1 \quad 2 \}$$

What is $x[n] \underset{N}{\circledast} y[n]$?

$w[n] = x[n] \underset{4}{\circledast} y[n]$

$$w[0] = 12$$
$$w[1] = 14$$
$$w[2] = 15$$
$$w[3] = 15$$
$$w[4] = w[0] = 12$$
$$w[5] = w[1] = 14$$
$$\vdots$$