Discrete convolution

And how to calculate it with the FFT

Herman Kamper

Discrete convolution

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

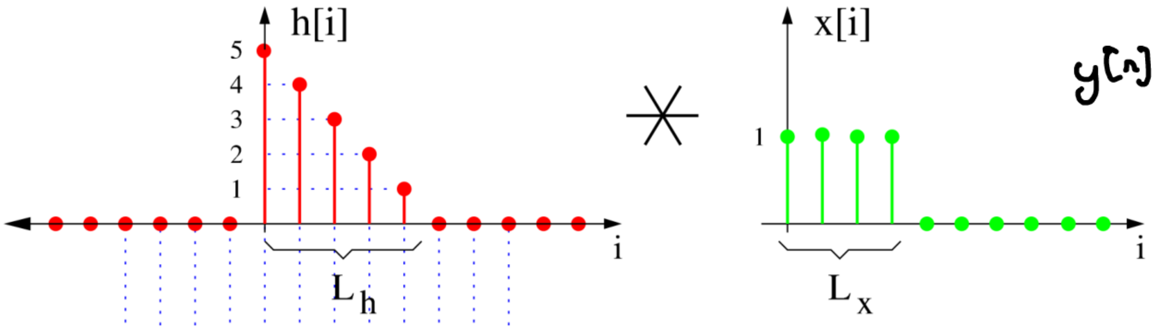
Definition:

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

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

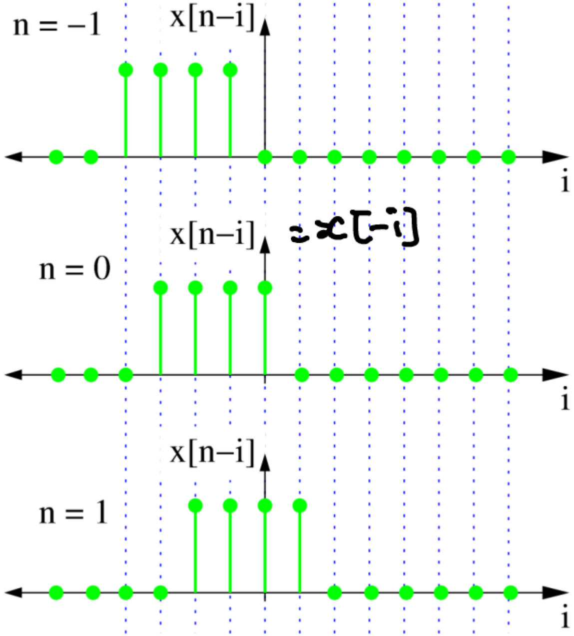


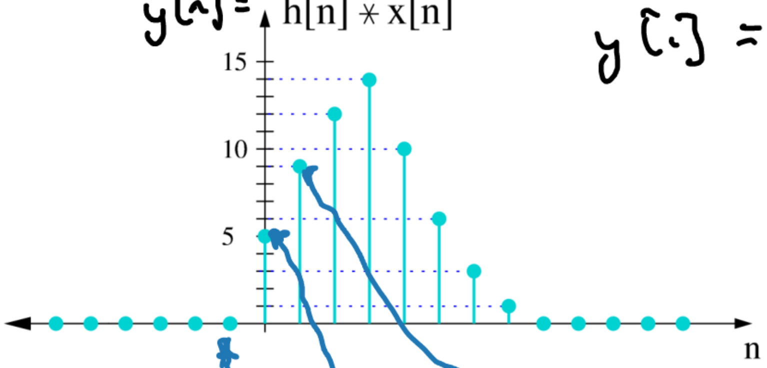
$$y[n] = h[n] * x[n] = \sum_{i=-\infty}^{\infty} h[i]x[n-i]$$

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

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

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

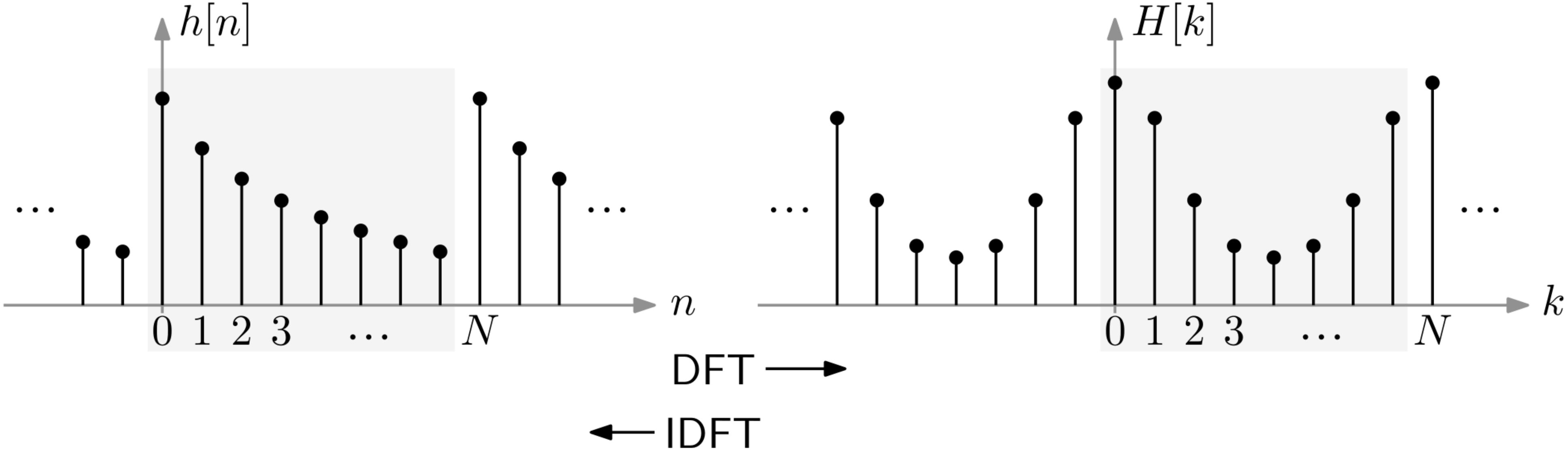




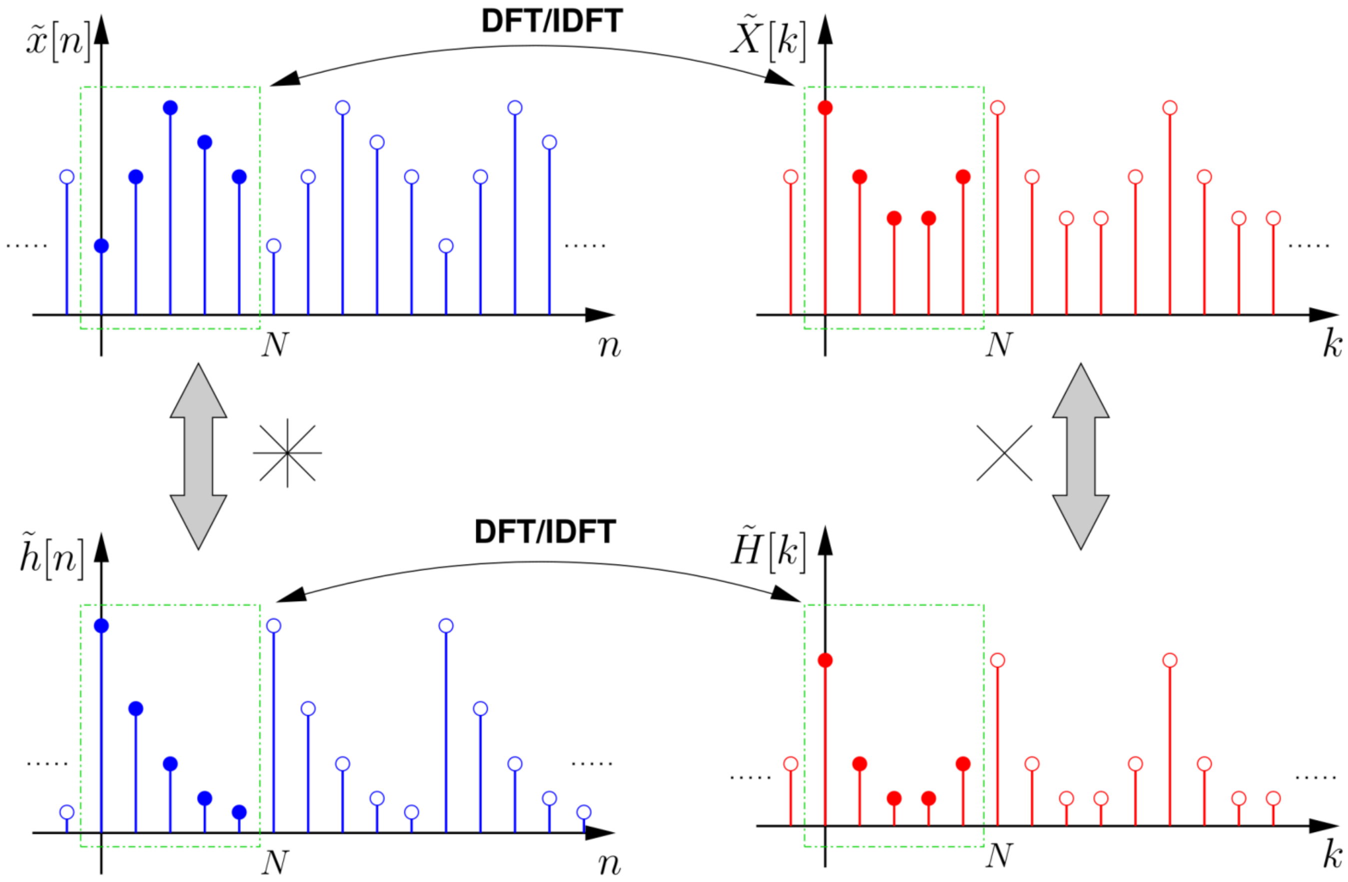
Why and how?

- Why this obsession with convolution? $x[n] \longrightarrow \boxed{h[n]} \longrightarrow x[n] * h[n]$
- Direct computation of discrete convolution takes a ton of ~~roughly~~ $\cancel{N^2}$ multiplications
- Can we do this using the FFT? $h(t) * x(t) \Longleftrightarrow H(f) \cdot X(f)$
  - 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 $N/2 \log_2 N$ 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 $\cdots$ N
$H[k]$
k
0 1 2 3 $\cdots$ N
DFT
IDFT

Discrete convolution and the DFT



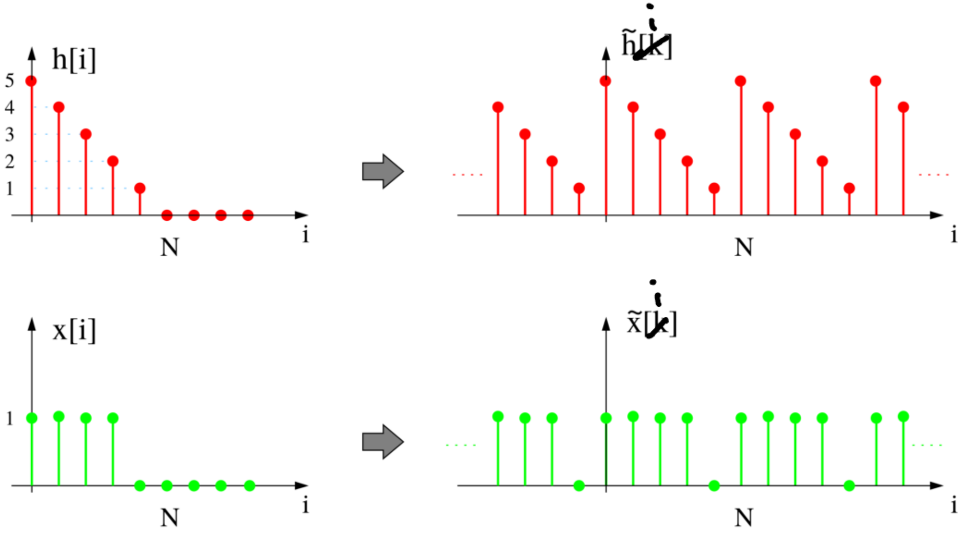
So what happens in time domain when we multiply DFTs?

Circular convolution:

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

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

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:

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

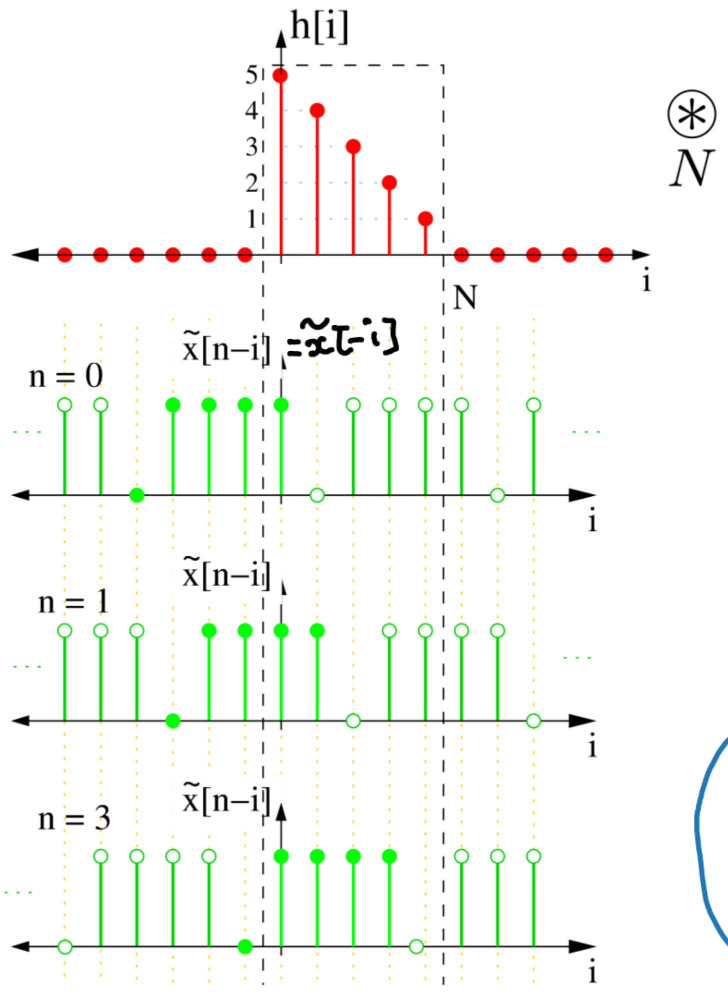
Proof that multiplying DFTs match circular convolution in time

Proakis and Manolakis (2007, Sec. 7.2.2)

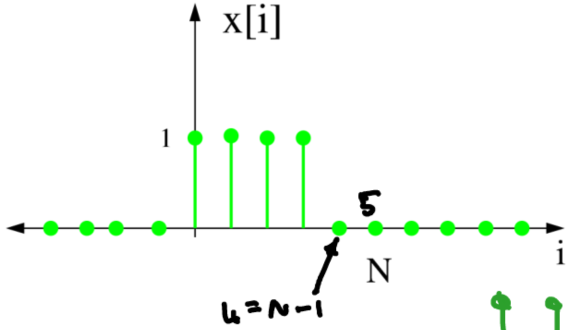
Circular convolution example

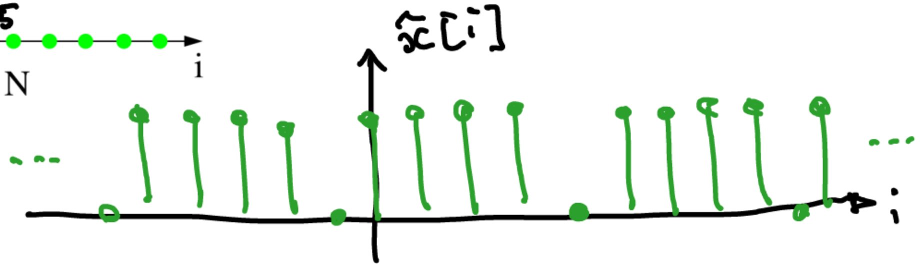
$$y[n] = h[n] \underset{N}{\circledast} x[n]$$

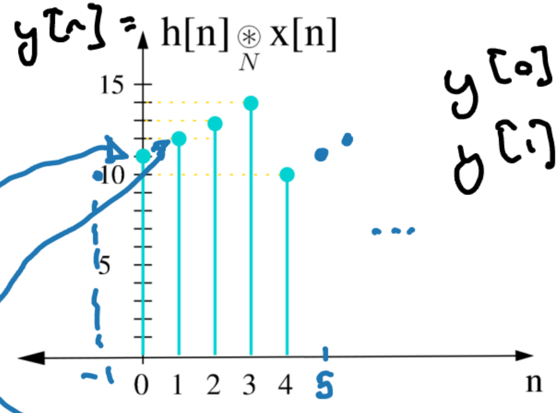
$$= \sum_{i=0}^{N-1} h[i] \cdot \tilde{x}[n-i]$$



y[0]:







$y[0] = 5 + 0 + 3 + 2 + 1 = 11$

$y[1] = 5 + 4 + 0 + 2 + 1 = 12$

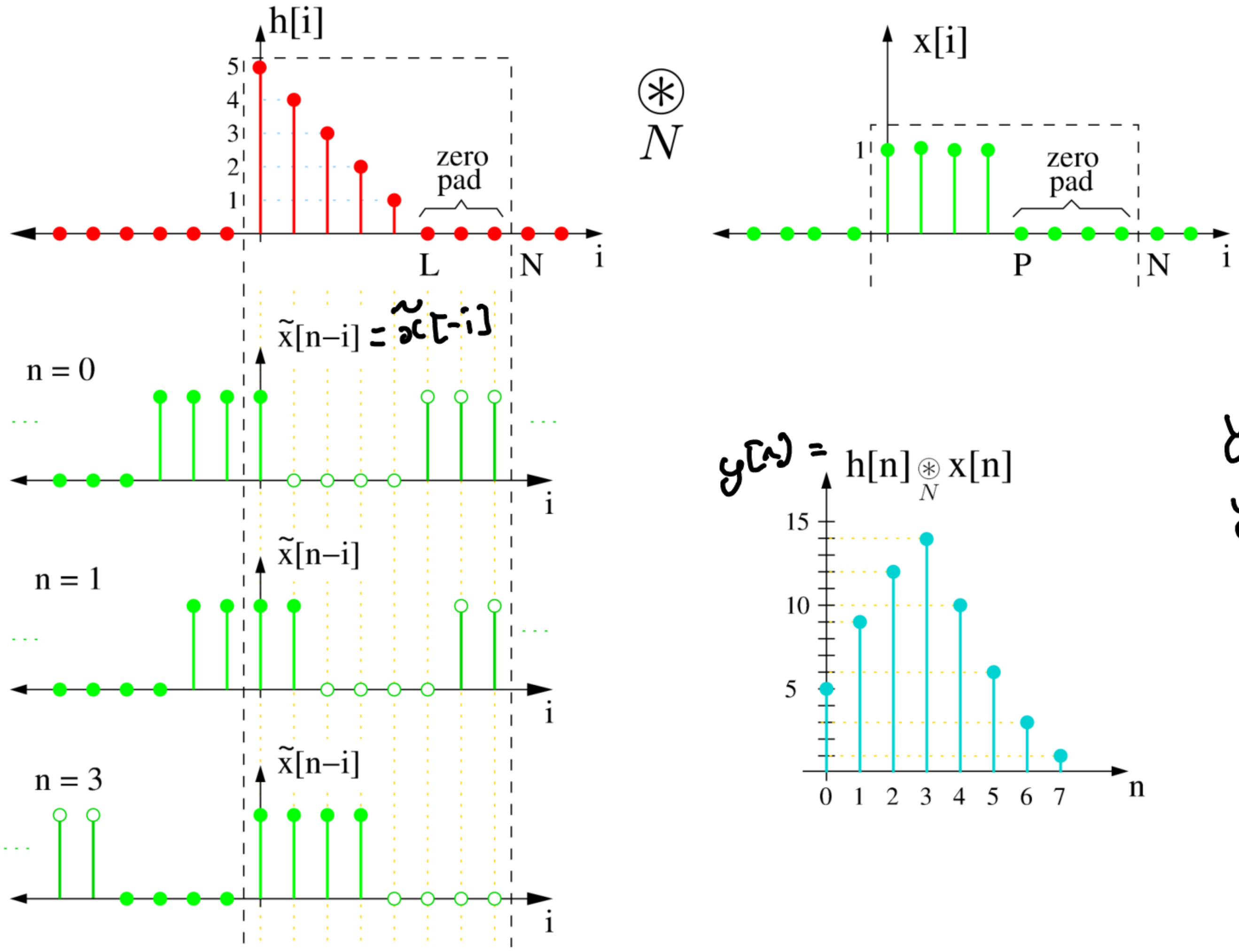
Zero padding: Discrete convolution via circular convolution

- 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

$y[0]$:

$y[1]$:



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

$y[0] = 5$

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

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

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

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

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

$w[0] = 12$

$w[1] = 14$

$w[2] = 15$

$w[3] = 15$

$w[4] = w[0] = 12$

$\vdots$