

4. Filtering Algorithms

Impulse response: $h_k := F((\delta_{j,0})_{j \in \mathbb{Z}}), (\dots, 0, 0, h_0, h_1, \dots, h_{m-1}, 0, \dots)$
 $= F((s_{j,z})_{j \in \mathbb{Z}}), (\dots, 0, 0, 0, h_0, h_1, \dots, h_{m-1}, 0, \dots)$

Discrete linear Convolution

$$\bullet \quad x_0 \cdot \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{m-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + x_1 \cdot \begin{bmatrix} 0 \\ h_0 \\ h_1 \\ \vdots \\ h_{m-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \dots + x_{n-1} \cdot \begin{bmatrix} 0 \\ \vdots \\ 0 \\ h_0 \\ h_1 \\ \vdots \\ h_{m-1} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n+m-1} \\ y_{n+m-2} \end{bmatrix}$$

- $x \ast h = y$
- $F((x_j)_{j \in \mathbb{Z}})_k = y_k = \sum_{j=0}^{n-1} x_j \cdot h_{k-j}$

Case $m=n$:

$$\bullet \quad \begin{bmatrix} h_0 & 0 & \dots & 0 \\ h_1 & h_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_{m-1} & h_1 & \dots & h_0 \\ 0 & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & h_{m-1} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_{n+m-2} \end{bmatrix}$$

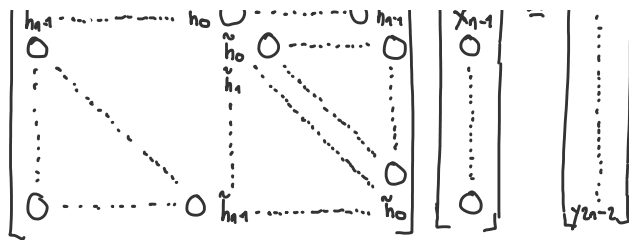
Maximal duration of output: $n+m-1$

Discrete circular/periodic convolution

In order that the dimensions work out we need to pad x and h to the length of y hence to a length $L := n+m-1$

- $y^L = x^L \ast_L p$
- $p_j = \sum_{\ell=0}^{\infty} h_{j+\ell n}$ (n -periodic)
 - $y_k = \sum_{j=0}^{n-1} p_j \cdot x_{k-j}$ ($y = p \ast_n x$)

$$\bullet \quad \begin{bmatrix} \tilde{h}_0 & 0 & \dots & 0 \\ \tilde{h}_1 & \tilde{h}_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{h}_{m-1} & \tilde{h}_1 & \dots & \tilde{h}_0 \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{h}_0 & \dots & 0 & \tilde{h}_{m-1} \end{bmatrix} \begin{bmatrix} \tilde{x}_0 \\ \vdots \\ \tilde{x}_{n-1} \end{bmatrix} = \begin{bmatrix} \tilde{y}_0 \\ \vdots \\ \tilde{y}_{n+m-2} \end{bmatrix}$$



Note: Linear convolution and periodic convolution give the same results. Just the way to get it is different. The reason we do periodic convolution is the computation. We can do it very fast with FFT.

Computation of Circular Convolution (Normal)

$$Y = X * h = \sum_{j=0}^{N-1} x_k h_j = F_N^{-1} \text{diag}(F_N p) F_N X = F_N^{-1} [(F_N X)_j (F_N h)_j]_{j=1}^N$$

- 1.) Zero Pad
- 2.) Compute their DFT's
- 3.) Pointwise multiplication
- 4.) Take Inverse DFT

$$F_N^{-1} = \frac{1}{N} \cdot F_N^H \quad O(N^2)$$

Complexity: Not faster than linear convolution. But now do FFT:

Computation of Circular Convolution (FFT)

$$\text{Complexity: } O\left(\frac{3N}{2} \cdot \log_2(N)\right) = O(N \cdot \log(N))$$

Eigen: 103, 110

Frequency Filtering with DFT

Densifying: Cutoff No. of Fourier coefficients

Trick: Only 1 high value in the Fourier spectrum \Rightarrow regular Graph

2D-DFT

$$U *_{m,n} X = \frac{1}{m \cdot n} \bar{F}_m \left[(F_m U F_n) \cdot (F_m X F_n) \right] \bar{F}_n \quad \bar{F}_n = F_n^H$$

Eigen: 5, 113