# ZS 4

Freitag, 11. Januar 2019 15:39

# 4. Filtering Algorithms

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

$:= F\big((\delta_{j,2})_{j\in\mathbb{Z}}\big)$, $(\ldots 0, 0, 0, h_0, h_1, \ldots, h_{m-1}, 0, \ldots)$

## Discrete linear Convolution

- $$x_0 \cdot \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{m-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + x_1 \begin{bmatrix} 0 \\ h_0 \\ h_1 \\ \vdots \\ h_{m-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \ldots + x_{n-1} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ h_0 \\ h_1 \\ \vdots \\ h_{m-1} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n+m-4} \\ y_{n+m-3} \\ y_{n+m-2} \end{bmatrix}$$
- $x * h = y$
- $F\big((x_j)_{j\in\mathbb{Z}}\big)_k = y_k = \sum_{j=0}^{n-1} x_j \cdot h_{k-j}$

**Case $m=n$:**

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

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 $L := n+m-1$

- $y^L = x^L *_L p$
- 1.) $p_j = \sum_{\ell=0}^{\infty} h_{j+\ell n}$ ($n$-periodic)

  2.) $y_k = \sum_{j=0}^{n-1} p_j \cdot x_{k-j}$ ($y = p *_n x$)

- $$\begin{bmatrix} \tilde{h}_0 & 0 & \cdots & 0 & \tilde{h}_{m-1} & \cdots & \tilde{h}_1 \\ \tilde{h}_1 & \ddots & & \vdots & 0 & \ddots & \tilde{h}_0 \\ \vdots & & \ddots & 0 & \vdots & & \vdots \\ \tilde{\ } & & & & & & \end{bmatrix} \begin{bmatrix} \tilde{x}_0 \\ \vdots \\ \tilde{\ } \end{bmatrix} = \begin{bmatrix} \tilde{y}_0 \\ \vdots \\ \vdots \end{bmatrix}$$

$$\begin{bmatrix} h_{n-1} & \cdots & h_0 & 0 & \cdots & 0 & h_{n-1} \\ 0 & & & \tilde{h}_0 & 0 & \cdots & 0 \\ \vdots & \ddots & & \tilde{h}_1 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \tilde{h}_{n-1} & \cdots & & \tilde{h}_0 \end{bmatrix} \begin{bmatrix} x_{n-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} \vdots \\ \vdots \\ y_{2n-2} \end{bmatrix}$$

**Note:** Linear convolution and perodic 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.

## Compution of Circular Convolution (Normal)

$$Y = x *_n h = \sum_{j=0}^{n-1} x_{k-j} h_j = F_n^{-1} \underbrace{\operatorname{diag}(F_n p')}_{} \underbrace{F_n x'}_{} = F_n^{-1} \left[ (F_n x)_j (F_n x)_j \right]_{j=1}^{n}$$

$$(\diagdown) \cdot (|)$$

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

$$F_n^{-1} = \frac{1}{n} \cdot F_n^H \qquad O(n^2)$$

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

## Compution of Circular Convolution (FFT)

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

Eigen: 109, 110

# Frequency Filtering with DFT

Denoising: Cutoff No. of Fourier coefficients

Trick: Only 1 high value in the Fourierspektrum ⇒ regular Graph

# 2D-DFT

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

Eigen: S. 119