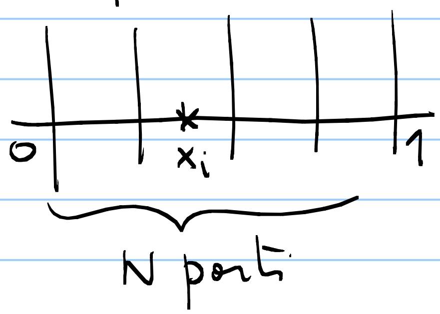
La FFT (Fast Fourier Transform)

Per comprendere la FFT dobbiamo ricorrere al formalismo dei numeri complessi.

DCT	DFT (Discrete Fourier Transform)
$\underline{y} = (y_0, \ldots, y_{N-1})$	$\underline{y} = (y_0, \ldots, y_{N-1})$ può essere in $\mathbb{C}$
	$\Rightarrow$ il prodotto scalare di vettori complessi diventa $\underline{z} \cdot \underline{y} = \sum_{j=0}^{N-1} z_j \overline{y_j}$ coniugato
	in modo che $\underline{z} \cdot \underline{z} = \sum z_j \overline{z_j} = \sum \lvert z_j \rvert^2 > 0$
$\underline{y} = \sum_{j=0}^{N-1} y_j \underline{e}_j$	$\underline{y} = \sum_{j=0} y_j \underline{e}_j$
	Gli $\underline{e}_j$ sono gli stessi di prima.

campionamenti:



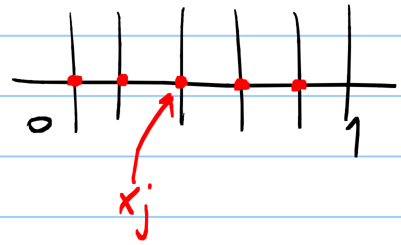
$$x_i = \frac{i}{N} + \frac{1}{2}\left(\frac{1}{N}\right) = \frac{2i+1}{2N}$$

Definiamo:

$$(\underline{w}_k)_j = \cos(k\pi x_j)$$

Per la DFT i campionamenti sono diversi:

$$x_j = \frac{j}{N}$$



Qui invece usiamo la serie intera:

$$(\underline{w}_k)_j = \cos(2\pi k x_j) + i\sin(2\pi k x_j)$$

$$= \cos\left(2\pi k \frac{j}{N}\right) + i\sin\left(2\pi k \frac{j}{N}\right)$$

$$= e^{i 2\pi k \frac{j}{N}}$$

che possiamo scrivere come

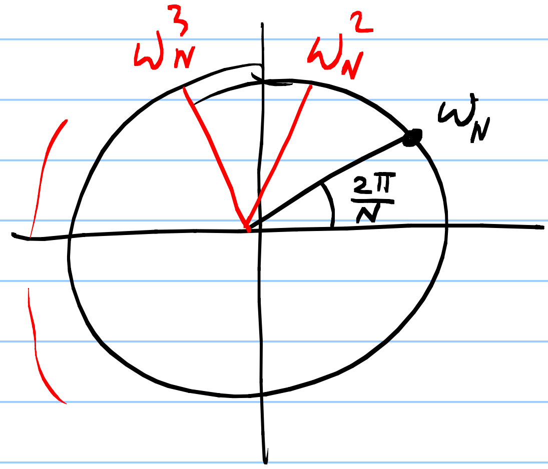
$$e^{i2\pi k \frac{j}{N}} = \left[e^{i\frac{2\pi}{N}}\right]^{kj}$$

Definiamo $\boxed{w_N = e^{i\frac{2\pi}{N}}}$

w_N è una radice N-esima dell'unità,

$$\boxed{w_N^N = 1}$$

w_N come numero complesso:



$$[\underline{w}_k]_j = \cos(k\pi x_j)$$

$$[\underline{w}_k]_j = \omega_N^{kj}$$

radice N-esima dell'unità

I $\underline{w}_k$ sono ortogonali

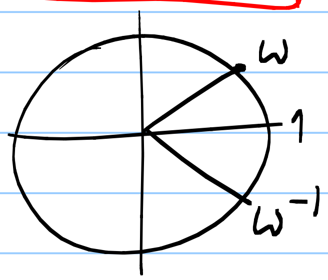
Anche i $\underline{w}_k$ sono ortogonali?

CONIUGATO

$$\underline{w}_k \cdot \underline{w}_\ell = \sum_{j=0}^{N-1} [\underline{w}_k]_j \overline{[\underline{w}_\ell]}_j =$$

$$= \sum_{j=0}^{N-1} \omega^{kj}\, \overline{\omega^{\ell j}} =$$

$$= \sum_{j=0}^{N-1} \omega^{kj}\, \overline{\omega}^{\ell j}$$



Ma $\boxed{\overline{\omega}_N = \frac{1}{\omega_N} = \omega_N^{-1}}$

quindi

$$\underline{w}_k \cdot \underline{w}_\ell = \sum_{j=0}^{N-1} \omega^{(k-\ell)j}$$

dobbiamo distinguere il caso $k=\ell$.

Se $\boxed{k=\ell}$:

$$\underline{w}_k \cdot \underline{w}_k = \sum_{j=0}^{N-1} 1 = N$$

da cui $\quad \|\underline{w}_k\| = \sqrt{N}$

Se $\boxed{k \neq \ell}$

$$\underline{w}_k \cdot \underline{w}_\ell = \sum_{j=0}^{N-1} \left[w^{k-\ell}\right]^j$$

io so che $1+s+\dots+s^{N-1} = \dfrac{s^N-1}{s-1}$

quindi

$$\underline{w}_k \cdot \underline{w}_\ell = \frac{\left[w^{(k-\ell)}\right]^N - 1}{w^{k-\ell} - 1} \quad *$$

Ma $\quad \left[w^{(k-\ell)}\right]^N = \left(\omega^N\right)^{k-\ell} = 1^{k-\ell} = 1$

e quindi $* = 0$.

L'ortogonalità si dimostra quindi facilmente.

Qui $\boxed{\text{NON}}$ facciamo la normalizzazione (ma non è importante).

$\underline{w}_k \rightsquigarrow \underline{\tilde{w}}_k$ normalizzati.

$\underline{y} = \sum y_i \underline{e}_i = \sum c_k \underline{\tilde{w}}_k$

$$y = \sum_{j=0}^{N-1} y_j \underline{e}_j = \sum_{k=0}^{N-1} c_k \underline{w}_k$$

Se moltiplichiamo $\cdot \underline{w}_\ell$:

CONIUGATO!

$$\sum_{j=0}^{N-1} y_j \overline{[\underline{w}_\ell]}_j = c_\ell \cdot N$$ non sono ortonormali!

$$\sum_{j=0}^{N-1} y_j \overline{w^{\ell j}} = c_\ell N$$

$$c_k = \alpha_k \sum_{j=0}^{N-1} y_j \cos(k\pi x_j)$$

DCT

$$\boxed{c_k = \frac{1}{N} \sum_{j=0}^{N-1} \overline{w}_N^{kj}} \qquad \boxed{w_N = e^{i\frac{2\pi}{N}}}$$

DFT

Trasformata inversa:

Moltiplichiamo per $\cdot \underline{e}_\ell$

$$y_\ell = \sum_{k=0}^{N-1} c_k [\underline{w}_k]_\ell =$$

$$= \sum_{k=0}^{N-1} c_k w_N^{k\ell}$$

$$y_j = \sum_{k=0}^{N-1} \alpha_k \cos(k\pi x_j)$$

$$\boxed{y_j = \sum_{k=0}^{N-1} c_k w_N^{kj}} \qquad \boxed{w_N = e^{i\frac{2\pi}{N}}}$$

IDCT

Indichiamo con F_N la matrice di componenti

$$(F_N)_{kj} = \omega_N^{kj}$$

e con $\vec{y}$, $\vec{c}$ i vettori

$\vec{y} = (y_0, \dots, y_{N-1})$ $\quad \vec{c} = (c_0, \dots, c_{N-1})$

DFT : $\quad \vec{c} = \frac{1}{N} \bar{F}_N \vec{y}$

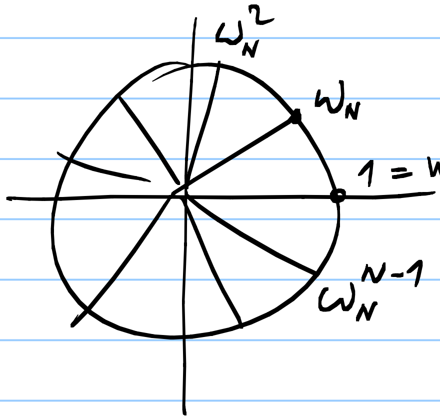
IDFT : $\quad \vec{y} = F_N \vec{c}$

Quindi $\quad \frac{1}{N} \bar{F}_N F_N = I$

da cui

$$\boxed{F_N^{-1} = \frac{1}{N} \bar{F}_N}$$

$$F = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w_N & w_N^2 & \cdots & w_N^{N-1} \\ 1 & w_N^2 & w_N^4 & \cdots & w_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w_N^{N-1} & w_N^{2(N-1)} & \cdots & w_N^{(N-1)^2} \end{bmatrix}$$



$\Rightarrow$ i w_N^k dopo un po' si ripetono!

Guardiamo la IDFT (la DFT è analoga).

$$\vec{y} = F_N \vec{c} \quad \rightarrow \quad O(N^2) \text{ operazioni a priori}$$

F_N: $N \times N$, $\vec{c}$: N

$$\boxed{(F_N)_{jk} = \omega_N^{jk}}$$

Vediamo che possiamo organizzare il calcolo in un modo diverso.

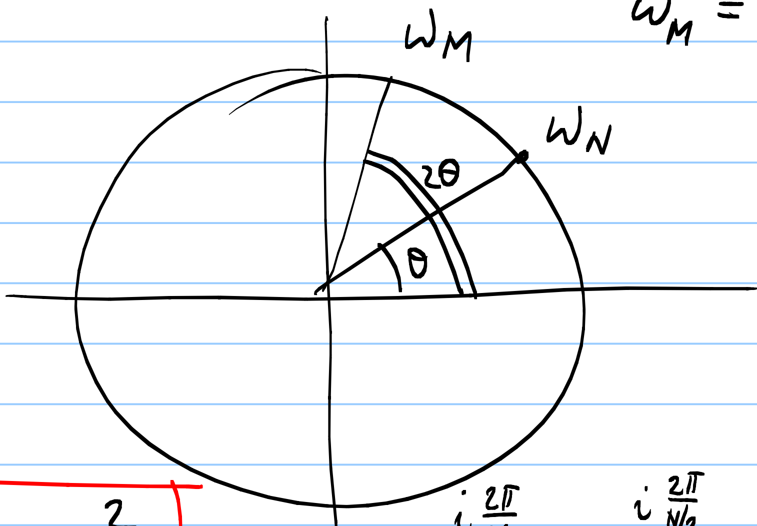
$$\vec{c} = (c_0, \ldots, c_{N-1}) \qquad (N \text{ PARI})$$

$$\vec{c}^{\,E} = (c_0, c_2, \ldots, c_{N-2}) \quad \text{PARI}$$

$$\vec{c}^{\,D} = (c_1, c_3, \ldots, c_{N-1}) \quad \text{DISPARI}$$

$\boxed{\text{Sia } M = \frac{N}{2}}$. Allora $\omega_N = e^{i\frac{2\pi}{N}}$ e

$$\omega_M = e^{i\frac{2\pi}{M}}$$



$$\boxed{\omega_M = \omega_N^2}$$

DIM. $e^{i\frac{2\pi}{M}} = e^{i\frac{2\pi}{N/2}} = e^{i\frac{4\pi}{N}} = \left(e^{i\frac{2\pi}{N}}\right)^2$

Allo stesso modo avremo che

F_N è la matrice $N \times N$ $(F_N)_{kj} = \omega_N^{kj}$ $\quad k=0,\dots,N-1$, $j=0,\dots,N-1$

F_M è la matrice $M \times M$ $(F_M)_{kj} = \omega_M^{kj}$ $\quad k=0,\dots,M-1$, $j=0,\dots,M-1$

$$\omega_M^{kj} = [\omega_N^2]^{kj} = \omega_N^{2kj}$$

Algoritmo:

$$\vec{c} < \begin{matrix} \vec{c}^{\,EVEN} \rightsquigarrow \vec{y}^{\,E} = F_M \vec{c}^{\,E} \\ \vec{c}^{\,ODD} \rightsquigarrow \vec{y}^{\,O} = F_M \vec{c}^{\,O} \end{matrix} > \to \text{DA CUI:}$$

$$\to \vec{y} = < \begin{matrix} y_j = y_j^E + \omega_N^j y_j^O & j=0,\dots,M-1 \\ y_{j+M} = y_j^E - \omega_N^j y_j^O & j=0,\dots,M-1 \end{matrix}$$

Dim

Partiamo da $y_j = \sum_{k=0}^{N-1} \omega_N^{kj} c_k$. (IDFT)

$\sum_{k=0}^{N-1} \omega_N^{kj} c_k =$ dividiamo in pari e dispari.

$$= \sum_{k=0}^{M-1} \omega_N^{(2k)j} c_{2k} + \sum_{k=0}^{M-1} \omega_N^{(2k+1)j} c_{2k+1} =$$

solo i pari $\qquad\qquad$ solo dispari

$$= \sum_{k=0}^{M-1} (\omega_N^2)^{kj} c_{2k} + \omega_N^j \sum_{k=0}^{M-1} (\omega_N^2)^{kj} c_{2k+1}$$

$$= \sum_{k=0}^{M-1} \omega_M^{kj} c_{2k} + \omega_N^j \sum_{k=0}^{M-1} \omega_M^{kj} c_{2k+1}$$

Abbiamo quindi l'identità:

$$y_j = \sum_{k=0}^{M-1} \omega_M^{kj} c_{2k} + \omega_N^j \sum_{k=0}^{M-1} \omega_M^{kj} c_{2k+1}$$

$j = 0, \dots, N-1$.

Se j va da 0 a $M-1$, allora possiamo scrivere

$$y_j = [F_M \vec{c}^E]_j + \omega_N^j [F_M \vec{c}^O]_j$$

Se j va da M a $N-1 = 2M-1$, al posto di j ci mettiamo $j+M$ e j va ancora da 0 a $M-1$:

$$y_{j+M} = \sum_{k=0}^{M-1} \omega_M^{k(j+M)} c_{2k} + \omega_N^{j+M} \sum_{k=0}^{M-1} \omega_M^{k(j+M)} c_{2k+1}$$

e $\omega_M^{k(j+M)} = \omega_M^{kj} \cdot \omega_M^{kM} = \omega_M^{kj} \cdot (\underbrace{\omega_M^M}_{=1})^k = \omega_M^{kj}$

$$\omega_N^{j+M} = \omega_N^j \omega_N^M$$

ma $\omega_N^M = \omega_N^{N/2} = \left[e^{i\frac{2\pi}{N}}\right]^{N/2} =$

$= e^{i\pi} = -1$ quindi

$$\omega_N^{j+M} = -\omega_N^j$$

e in definitiva

$$y_{j+M} = \sum_{k=0}^{M-1} \omega_M^{kj} c_{2k} - \omega_N^j \sum_{k=0}^{M-1} \omega_M^{kj} c_{2k+1}$$

$$= \left(F_M \vec{c}^{\,E}\right)_j - \omega_N^j \left(F_M \vec{c}^{\,O}\right)_j$$

Quindi:

$$y_j = \left[F_M \vec{c}^{\,E}\right]_j + \omega_N^j \left[F_M \vec{c}^{\,O}\right]_j \quad j=0,\dots,M-1$$

$$y_{j+M} = \left(F_M \vec{c}^{\,E}\right)_j - \omega_N^j \left(F_M \vec{c}^{\,O}\right)_j \quad j=0,\dots,M-1$$

Dobbiamo quindi effettuare 2 DFT di dimensione $N/2$ più N molt. per ω_N^j;

$\boxed{N = 2^P}$

$\mathrm{DFT}(N) \approx 2\,\mathrm{DFT}\left(\frac{N}{2}\right) + N$ iterando:

$$\approx 2\left[2\,\mathrm{DFT}\left(\frac{N}{4}\right) + \frac{N}{2}\right] + N$$

$$\vdots$$

$$\approx 2^P\,\mathrm{DFT}(1) + \underbrace{N + \dots + N}_{p \text{ volte}} \approx$$

$$\approx N\,DFT(1) + pN =$$

$$\approx N \cdot DFT(1) + N \log_2 N$$

$$\approx N \log_2 N.$$

Quindi si vede che la complessità si abbassa da N^2 a $N \log_2 N$.

Ecco cosa scrive G. Strang a pag. 449 di "Introduction to Applied Mathematics":

Our interest is in powers like $n = 2^{12}$. There will be $n^2 = 2^{24}$ entries in F_n, and an ordinary matrix-vector product $F_n x$ requires 2^{24} complex multiplications. In itself that is not terrible; it takes a few seconds on a big machine.† But if it is repeated thousands of times, as it is in time series analysis and image processing and elsewhere, the cost of these products $F_n x$ becomes prohibitive. By contrast, the Fast Fourier Transform finds $F_n x$ with only $6 \cdot 2^{12}$ multiplications—it is *more than* 600 *times faster*. It replaces n^2 multiplications by $\frac{1}{2}nl$, when $n = 2^l$. Thus $l = \log_2 n$ and the factor $n = 2^{12}$ is exchanged for $\frac{1}{2}l = 6$. By connecting F_n to $F_{n/2}$, and then to $F_{n/4}$, and eventually to F_1, the usual n^2 steps are reduced to $\frac{1}{2}n \log_2 n$. Practically speaking, we have n instead of n^2.