Discrete Fourier Transform.

Requirement: used for signal that are periodic and discrete in t-domain.

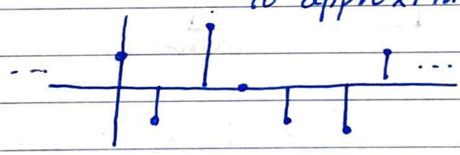
In DFT, we have
- DFT:
- DTFS: DT Fourier Series
- FFT: Fast Fourier Transform.

Intuition:

The $\boxed{F.S.}$ takes a continuous, periodic t-signal and Represents it as a sum of complex sin's & cosine's.

$$x(t) = \sum_{k=-\infty}^{\infty} \underbrace{X[k]}_{a_k} e^{jk\overbrace{\frac{2\pi}{T}}^{\omega_0}t} \qquad t \in [0, T]$$

you need infinitely many a_k to approximate $x(t)$.

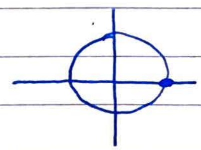


Could we then write: $N=6 \leftarrow$ period.

$$x(n) = \sum_{k=-\infty}^{\infty} X(k) e^{jk\frac{2\pi}{N}n} \qquad n = 0, \dots, N-1$$

D.T. ↖ finite infinite.

$$e^{jk\frac{2\pi}{N}(n+N)} = e^{jk\frac{2\pi}{N}n} \underbrace{e^{jk2\pi}}_{=1}$$



$$= e^{jk\frac{2\pi}{N}n}$$

There are only N complex unique exponentials of period N.

Define the discrete time fourier transform (DFT):

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-jk\frac{2\pi}{N}n} \qquad k = 0, \dots, N-1$$

$$x[n] = \frac{1}{N} \sum_{n=0}^{N-1} X[k] e^{+jk\frac{2\pi}{N}n} \qquad n = 0, \dots, N-1$$

(*) This here gives us how to go from N number of x[n] to N number of value in DFT.

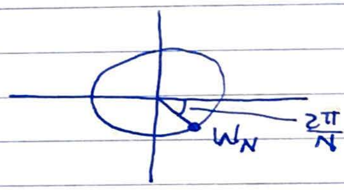
To make notation more easier to write:

We say $W_N^{kn} = e^{-jk\frac{2\pi}{N}n}$

$$\boxed{W_N = e^{-j\frac{2\pi}{N}}}$$

↳ nth Root of 1

$$\boxed{(W_N)^N = 1}$$



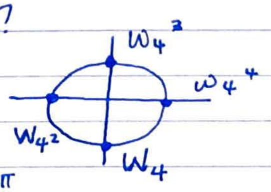
Now with the new notation:

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}$$

$(W_2)^1 = e^{-j\frac{2\pi}{2}} = e^{-\pi} = -1$

$(W_2)^2 = e^{-2\pi} = 1$

What about W_4?



The 4 Root of 1

$Z^4 = 1$

$e^{-2\pi j}$ $\quad W_4 = e^{-j\frac{2\pi}{4}}$

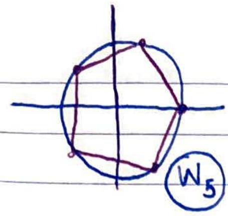
$e^{-4\pi j}$ $\quad (W_4)^2 = e^{-j\frac{\pi}{2}\cdot 2} = e^{-j\pi}$

$e^{-6\pi j}$ $\quad (W_4)^3 = e^{-j\frac{\pi}{2}\cdot 3} = e^{-j\frac{\pi 3}{2}}$

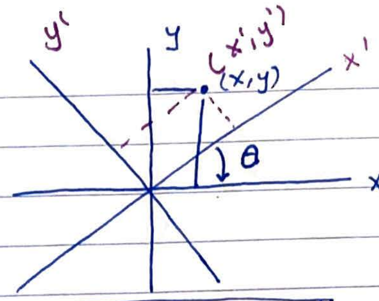
$e^{-8\pi j}$ $\quad (W_4)^4 = e^{-j\frac{\pi}{2}\cdot 4} = e^{-j2\pi} = e^{j0}$

$(W_N)^N = 1$ $\quad W_N \quad (W_N)^{\frac{N}{2}}$

$(W_N)^N$



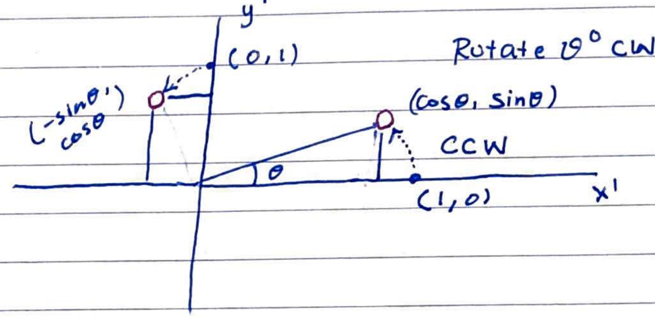
In Relate to
Linear Algebra.



In machine Learning, we use $\boxed{\text{PCA analysis}}$ to transform (x, y) into the co-ordinate of the sys. where x' = most variation of data, y' = least variation of data.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

↑ new co-ordinate ↑ ↑ original data.



Dot Product of the operation matrix is ZERO → orthogonal

We want to change old c.p. w/ ⊥ to c.p.' w/ ⊥ new co-ord.

So: $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} a \\ c \end{bmatrix}$

So think of $X[k] = \sum_n x[n] W_N^{kn}$ as:

$$\begin{bmatrix} X[0] \\ X[1] \\ \vdots \\ X[N-1] \end{bmatrix} = \begin{array}{l} n=0 \\ n=1 \\ n=2 \\ n=N-1 \end{array} \begin{bmatrix} W_N^0 & W_N^0 & W_N^0 \cdots & W_N^0 \\ W_N^0 & W_N^1 & \cdots\cdots & W_N^{k-1} \\ W_N^0 & W_N^2 & \cdots\cdots & W_N^{2k(N-1)} \\ W_N^0 & W_N^{N-1} & \cdots, \cdots, & W_N^{(N-1)^2} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}$$

$\quad k=0 \quad k=1 \quad k=N-1$

output $N \times 1$ complex. operation $N \times N$ complex matrix. input $N \times 1$.

$X[k] = F\, x[n]$ ← output; F: DFT; x[n]: input.

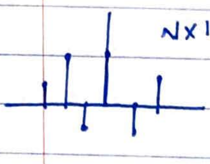
F is orthogonal, colums all length N. are all ⊥.

DFT : co-ordinate transformation to f-domain.

$$[\ \ ] = X[0]\,[\ \ ] + X[1]\,[\ \ ] + \cdots$$

↑ basis vectors ↗

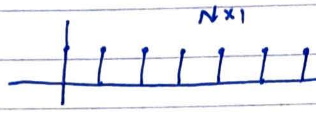
N×1



$=$ X [0]

N

N×1



you can quickly do a Length-4 DFT → pretty easy matrix.

$$\text{DFTMTX}(2) \rightarrow \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Well, we wait to simplify this process.....

How is this Related to DTFT?

Say we have a finite length discrete-time signal where

$$\{x[0], x[1], \cdots, x[N-1]\} \neq 0$$

DTFT

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n} \qquad \omega \in [-\pi, \pi]$$

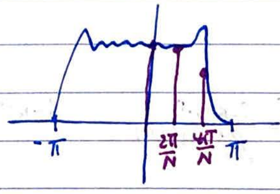
continuous ↗

$$= \sum_{n=0}^{N-1} x[n] e^{-j\omega n}$$

if I were to evaluate ω where $\boxed{\omega = \frac{2\pi k}{N}}$

DFT

$$X[k] = X\left(\frac{2\pi k}{N}\right)$$



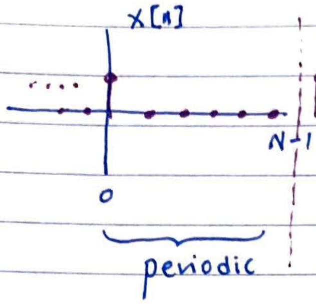
Insight

If I want to sample the $X(\omega)$ ... do many N!

DFT is the DTFT of the finite length signal evaluated at N equally spaced points $\omega = \frac{2\pi k}{N}$, $k = 0, \cdots, N-1$

MATLAB Example - Pending ...

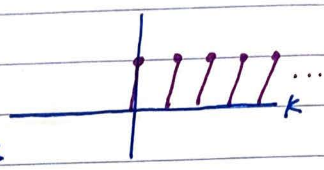
Example of DFTs:



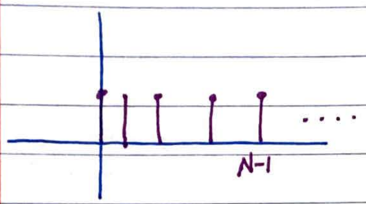
$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk}$$

$$= W_N^{0k}$$

$$= 1 \quad \forall k$$



$\delta[n]$



$$X[k] = \sum_{n=0}^{N-1} W_N^{nk} \quad \text{geometric Series.}$$

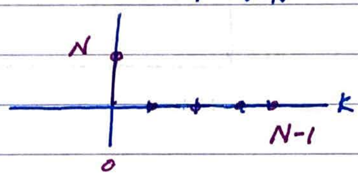
$$= \sum_{n=0}^{N-1} (W_N^{k})^n$$

$$= \frac{1 - W_N^{k(N)}}{1 - W_N^{k}}$$

if $k=0$, then $X[k] = \sum_{n=0}^{N-1} 1^n = N$ $\qquad e^{-j\frac{2\pi}{N}\cdot kN} = e^{-j2\pi} = 1$

if $k \neq 0$, then $X[k] = \sum_{n=0}^{N-1} (W_N^{k})^n = \frac{1 - W_N^{kN}}{1 - W_N^{k}} = \frac{1 - 1}{\cdots} = 0$

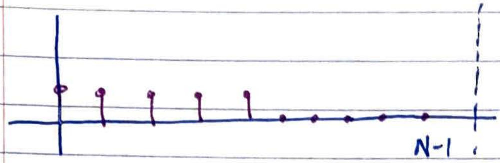
And so: $X[k] = \begin{cases} N & k=0 \\ 0 & k \neq 0 \end{cases}$



Orthogonality Property.

$$\sum_{n=0}^{N-1} W_N^{Mn} = \begin{cases} N, & M \text{ is an integer multiple of } N \\ 0, & \text{else.} \end{cases}$$

Another example.



For DFT case:

$\boxed{N=10}$

DTFT

$$X(W) = \sum_{n=0}^{4} e^{-j\omega n} = \sum_{n=0}^{4} (e^{-j\omega})^n = \frac{1-e^{-5j\omega}}{1-e^{-j\omega}}$$

$$\frac{1-e^{-5j\omega}}{1-e^{-j\omega}} = \frac{e^{-5j\frac{\omega}{2}}}{e^{j\frac{\omega}{2}}} \frac{e^{5j\frac{\omega}{2}} - e^{-5j\frac{\omega}{2}}}{e^{j\omega/2} - e^{-j\omega/2}} = e^{-2j\omega} \frac{2j\sin(\frac{5\omega}{2})}{2j\sin(\omega/2)} = \boxed{e^{-2j\omega} \frac{\sin\frac{5\omega}{2}}{\sin\frac{\omega}{2}}}$$

DFT:

$N=10$

$N-1=9$

Let's eval. DTFT. at $W = \frac{2\pi K}{10} \rightarrow e^{-2j(\frac{2\pi K}{10})} \frac{\sin(\frac{5}{2}(\frac{2\pi K}{10}))}{\sin(2\pi K/10 \cdot \frac{1}{2})}$

$= \text{DFT}\{x(n)\}$

$$X[K] = \sum_{n=0}^{9} x[n] W_{10}^{kn}$$

$$= \sum_{n=0}^{4} W_{10}^{kn} = \sum_{n=0}^{4} (W_{10}^{k})^n = \frac{1-W_{10}^{k5}}{1-W_{10}^{k}} = \frac{1-e^{-j\frac{2\pi}{10}k5}}{1-W_{10}^{k}}$$

$$= \frac{e^{-jk\frac{\pi}{10}}(e^{jk\frac{\pi}{10}} - e^{-jk\frac{9\pi}{10}})}{e^{-jk\frac{\pi}{10}}(e^{+jk\frac{\pi}{10}} - e^{-jk\pi/10})} = \frac{e^{-jk\frac{4\pi}{10}}(e^{jk\frac{5\pi}{10}} - e^{-jk\frac{5\pi}{10}})}{\sin(k\frac{\pi}{10})}$$

$$\boxed{= e^{-jk4\frac{\pi}{10}} \frac{\sin(k\frac{\pi}{2})}{\sin(k\frac{\pi}{10})}}$$

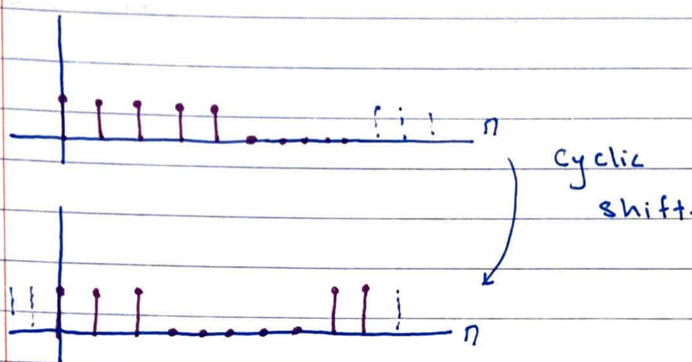
$\forall K \in [0, \cdots, 9]$

Properties: DFT.

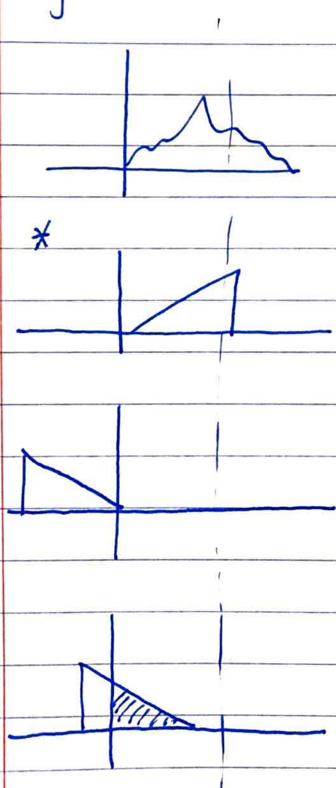
- Linearity, Symmetry.

- shift

$$x[n-m] \overset{DFT}{\longleftrightarrow} W_N^{km} \, X[k]$$

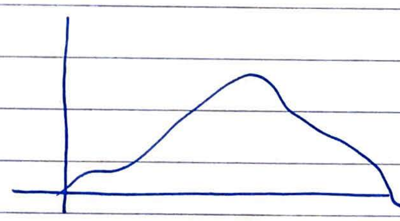


Cyclic Convolution. (⊛)

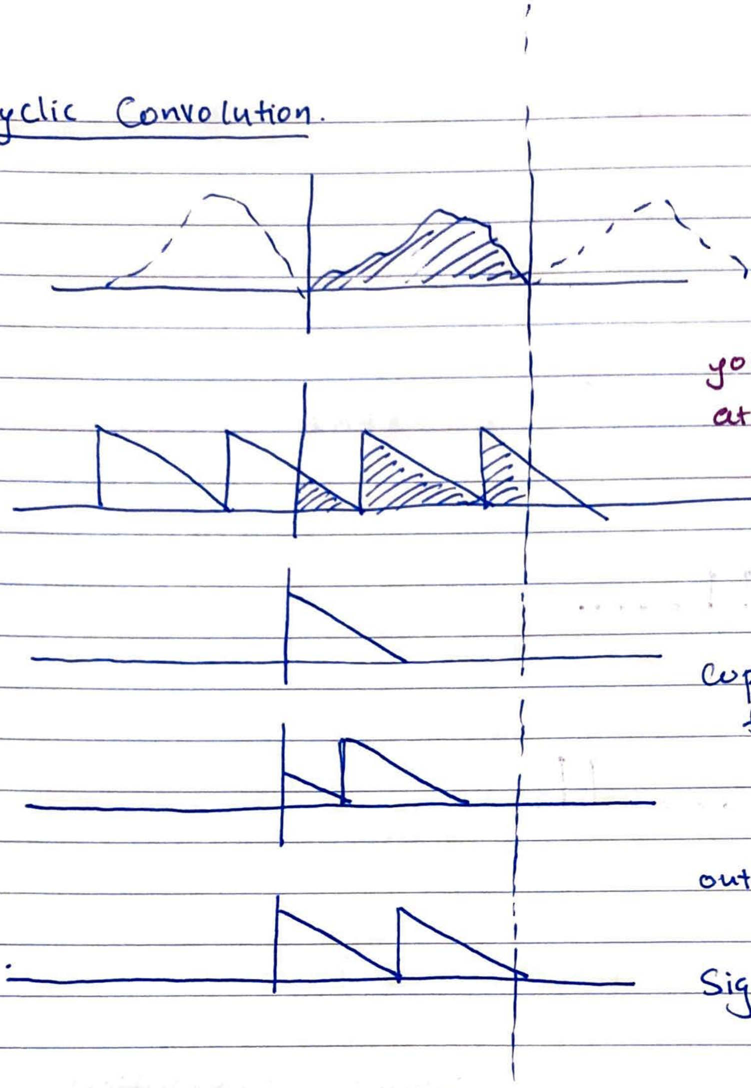


Linear Convolution

Regular Conv.



Cyclic Convolution.



you shift and look at overlap bump by bump.

copies coming in from the left

↓

output is $\boxed{\text{periodic}}$

Signal wraps!

For the DFT, we have the same kind of convolution multiplication property but they're cyclic.

$$\underset{\uparrow}{x[n]} \circledast \underset{\uparrow}{h[n]} \xleftrightarrow{\text{DFT}} X[k]H[k]$$

Length N

But, how to get "Regular" Convolution we need for LTI System?

Suppose x and h are length-N signal. What is $x \circledast h$?

$x[0], x[1], \cdots x[N-1] \neq 0$

$$\begin{array}{|cccc|c}
x[0] & x[1] & \cdots & x[N-1] & y[0] \\
h[0] & h[N-1] & \cdots & h[1] & \\
h[1] & h[0] & \cdots & h[2] & y[1]. \\
h[2] & h[1] & \cdots & h[3] & y[2] \\
\vdots & \vdots & & \vdots &
\end{array}$$

$y[0] = h[0]\,x[0] + h[N-1]\,x[1] + \cdots + h[1]\,x[N-1]$
$y[1] = h[1]\,x[0] + h[0]\,x[1] + \cdots + h[2]\,x[N-1]$
$y[N-1] = h[N-1]\,x[0] + h[1]\,x[1] + \cdots + h[0]\,x[N-1]$

Simplify:

$$\underbrace{\begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[N-1] \end{bmatrix}}_{N\times 1} = \underbrace{\begin{bmatrix} h[0] & h[N-1] & & h[1] \\ h[1] & h[0] & & h[2] \\ & & \ddots & \\ h[N-1] & h[N-2] & & h[0] \end{bmatrix}}_{N\times N} \underbrace{\begin{bmatrix} x[0] \\ \\ \\ x[N-1] \end{bmatrix}}_{N\times 1}$$

Circulant matrix.

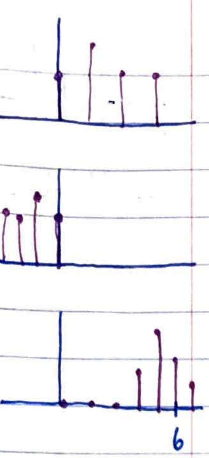
This matrix product is calculating the circular Convolution.

Consider just Linear Convolution:

$x[0] \cdots x[3]$
$h[0] \cdots h[3]$

$y[0] = x[0]\,h[0]$
$y[1] = x[0]\,h[1] + x[1]\,h[0]$

$m+n-1 = 7$



Think about matrix product!

$$\begin{bmatrix} y[0] \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ y[6] \end{bmatrix} = \begin{bmatrix} h[0] & 0 & 0 & 0 \\ h[1] & h[0] & 0 & 0 \\ h[2] & h[1] & h[0] & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & h[3] \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}$$

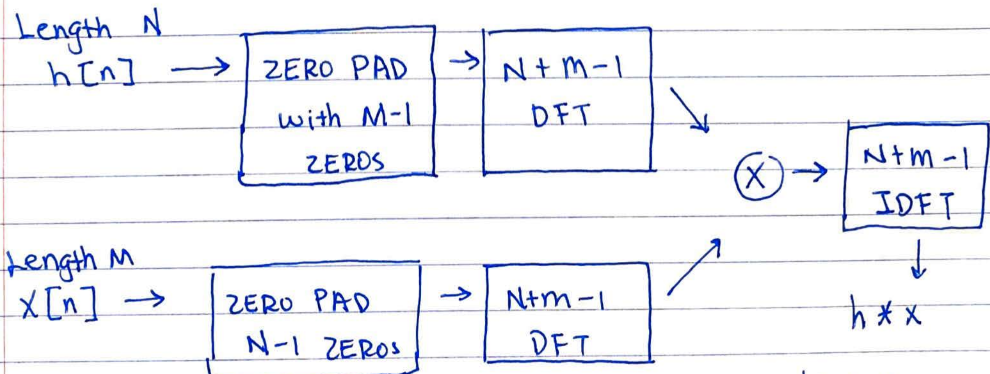
7×1 $\quad$ 7×4 $\quad$ 4×1

↑ no wrapping and not square

we like matching dimension

$$\begin{bmatrix} y[0] \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ y[6] \end{bmatrix} = \left[\begin{array}{cccc|ccc} h[0] & 0 & 0 & 0 & h[3] & h[2] & h[1] \\ h[1] & h[0] & 0 & 0 & & h[3] & h[2] \\ h[2] & & h[0] & & & & h[3] \\ h[3] & & & h[0] & & & \\ & & & & h[0] & & \\ & & & & & h[0] & \\ 0 & 0 & 0 & h[3] & h[2] & h[1] & h[0] \end{array}\right] \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

To do linear Convolution with DFT. Here's procedure



Linear convolution.

100 $\ast$ 49

↓ $\quad$ ↓

149 $\quad$ 149.