Fast 1D Convolution Algorithms summary

P. Giannakeris, P. Bassia, N. Kilis, R.T. Chadoulis
Prof. Ioannis Pitas
Aristotle University of Thessaloniki
pitas@csd.auth.gr
www.aiia.csd.auth.gr

Version 4.1.1

Artificial Intelligence &
Information Analysis Lab

Fast 1D Convolution Algorithms

- **Convolution Algorithms**
- Linear Convolutions
- Winograd Linear Convolution
- Cyclic Convolutions
- 1D FFT
- Winograd Cyclic Convolution
- Nested convolutions
- Block convolutions
- Applications
    - Convolutional neural networks.

Convolution Algorithms

- Machine learning
  - **Fast implementation of 1D/2D/3D convolutions in**
  - **Convolutional Neural Networks (CNNs).**
- Fast implementation of 1D digital filters
  - 1D signal filtering (e.g., audio/music, ECG, EEG)
  - 1D Signal feature calculation
- Fast implementation of 1D correlation
  - 1D template matching
  - Time-of-flight (distance) calculation (e.g., sonar)

Convolution Algorithms

- Fast implementation of 2D/3D convolutions:
  - Image/video filtering
  - Image/video feature calculation:
    - Gabor filters
    - Spatiotemporal feature calculation

- Fast implementation of 2D correlation:
  - Template matching
  - Correlation tracking.

Fast 1D Convolution Algorithms

- Convolution Algorithms
- **Linear Convolutions**
- Winograd Linear Convolution
- Cyclic Convolutions
- 1D FFT
- Winograd Cyclic Convolution
- Nested convolutions
- Block convolutions
- Applications
    - Convolutional neural networks.

Linear 1D convolution

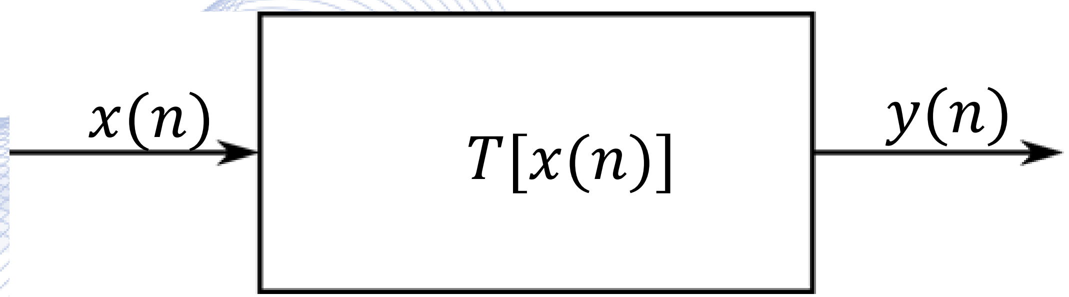
- Linearity:

$$T[ax_1 + bx_2] = aT[x_1] + bT[x_2].$$

- Shift-Invariance:

$$y(n) = T[x(n)] \Rightarrow$$

$$y(n-m) = T[x(n-m)].$$



LSI system convolution : $y(n) = h(n) * x(n).$

Linear 1D convolution

The one-dimensional (linear) convolution of:

- an input signal x of length L and
- a convolution kernel h (filter mask, finite impulse response) of length M is defined as:

$$y(n) = h(n) * x(n) \triangleq \sum_{i=0}^{M-1} h(i)x(n-i).$$

- For a convolution kernel centered around 0 and $M = 2v + 1$, convolution takes the form:

$$y(n) = h(n) * x(n) = \sum_{i=-v}^{v} h(i)x(n-i).$$

Linear 1D convolution

Vectorial convolution input/output, kernel representation:

- $\mathbf{x} = [x(0), \dots, x(L-1)]^T$: input vector.
- $\mathbf{h} = [h(0), \dots, h(M-1)]^T$: filter corfficient vector.
- $\mathbf{y} = [y(0), \dots, y(N-1)]^T$: output vector, with $N = L + M - 1$.

- 1D linear convolution between two discrete signals $\mathbf{x}, \mathbf{h}$ can be expressed as the matrix-vector product:

$$\mathbf{y} = \mathbf{H}\mathbf{x},$$

where $\mathbf{H}$ is a $N \times L$ matrix.

Linear 1D convolution

- $\mathbf{H}$: a $N \times L$ band matrix of the form:

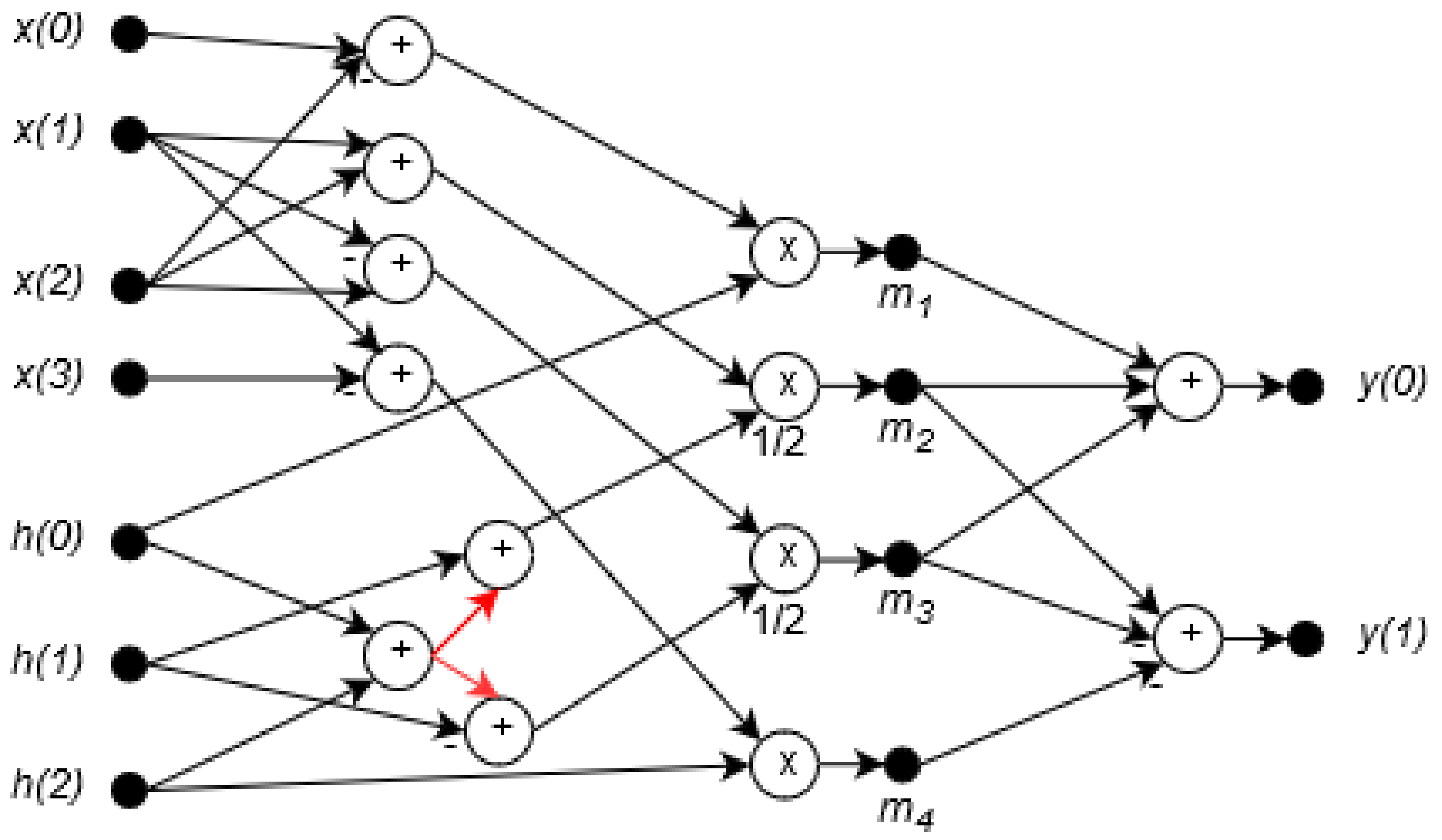
$$\mathbf{H} = \begin{bmatrix} h(0) & 0 & \cdots & 0 \\ h(1) & h(0) & \cdots & \cdots \\ \cdots & \cdots & \cdots & 0 \\ h(M-1) & h(M-2) & \cdots & 0 \\ 0 & h(M-1) & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & h(M-1) \end{bmatrix}.$$

- Alternative matrix notation: $\mathbf{y} = \mathbf{X}\mathbf{h}$, where $\mathbf{X}$ is an $\mathrm{N} \times M$ matrix.
- Fast calculation of the product $\mathbf{y} = \mathbf{H}\mathbf{x}$ using BLAS/cuBLAS.

Fast 1D Convolution Algorithms

- Convolution Algorithms
- Linear Convolutions
- **Winograd Linear Convolution**
- Cyclic Convolutions
- 1D FFT
- Winograd Cyclic Convolution
- Nested convolutions
- Block convolutions
- Applications
  - Convolutional neural networks.

Winograd Linear Convolution



Winograd linear convolution: Intermediate addition result is used 2 times.

Winograd Linear Convolution

- Winograd linear convolution algorithm requires $m + r - 1$ multiplications, m and r: lengths of y and h, respectively.

- General form of optimal Winograd linear convolution algorithms:

$$\mathbf{y} = \mathbf{A}^T[(\mathbf{Hh}) \otimes (\mathbf{B}^T\mathbf{x})],$$

- $\otimes$ indicates element-wise $m + r - 1$ multiplications.
- $\mathbf{x}, \mathbf{h}, \mathbf{y}$: input signal, filter coefficient and output signal vectors.

Fast 1D Convolution Algorithms

- Convolution Algorithms
- Linear Convolutions
- Winograd Linear Convolution
- **Cyclic Convolutions**
- 1D FFT
- Winograd Cyclic Convolution
- Nested convolutions
- Block convolutions
- Applications
  - Convolutional neural networks.

Cyclic 1D convolution

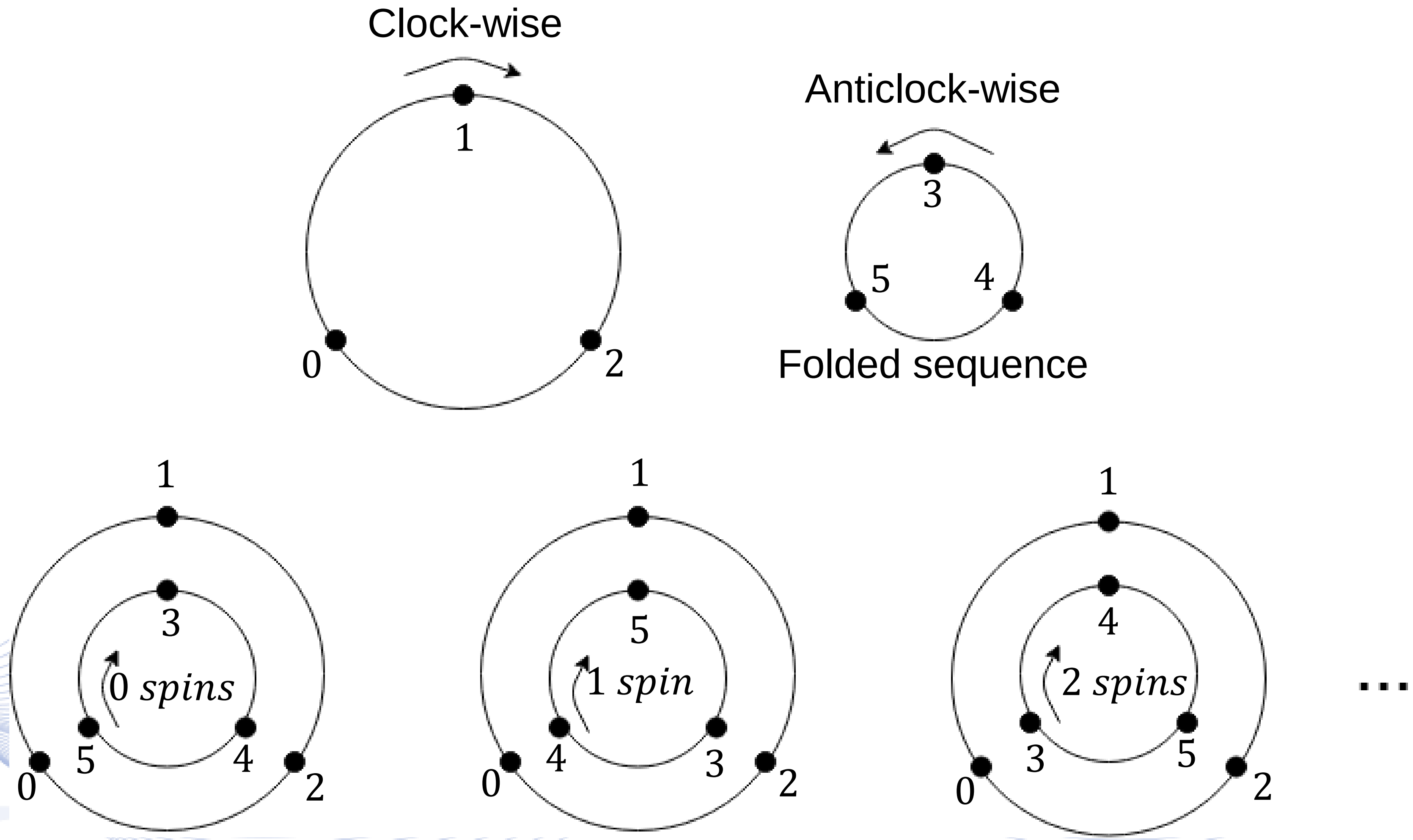
- One-dimensional cyclic convolution of length N :

$$y(k) = x(k) \circledast h(k) = \sum_{i=0}^{N-1} h(i)x\big(((k-i)_N)\big),$$

$$(k)_N = k \mod N.$$

- It is of no use in modeling linear systems.
- Important use: Embedding linear convolution in a **fast** cyclic convolution $y(n) = x(n) \circledast h(n)$ of length $N \geq L + M - 1$ and then performing a cyclic convolution of length N.

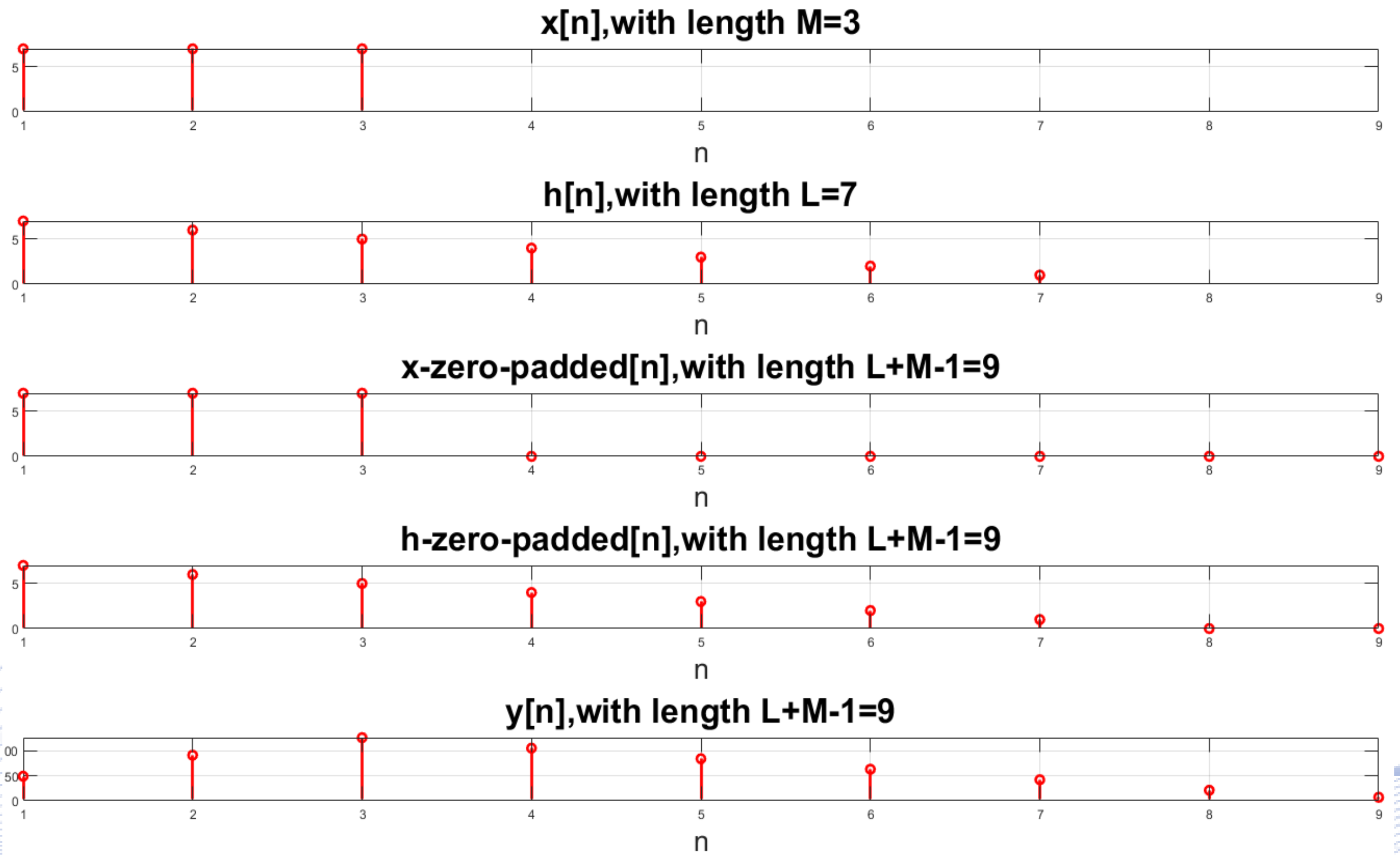
Cyclic 1D convolution



$y(0) = 1 \times 3 + 2 \times 4 + 0 \times 5$ $\quad y(1) = 1 \times 5 + 2 \times 3 + 0 \times 4$ $\quad y(2) = 1 \times 4 + 2 \times 5 + 0 \times 3$

Cyclic convolution of $x(n) = \{1, 2, 0\}$ and $h(n) = \{3, 5, 4\}$.

Cyclic 1D convolution

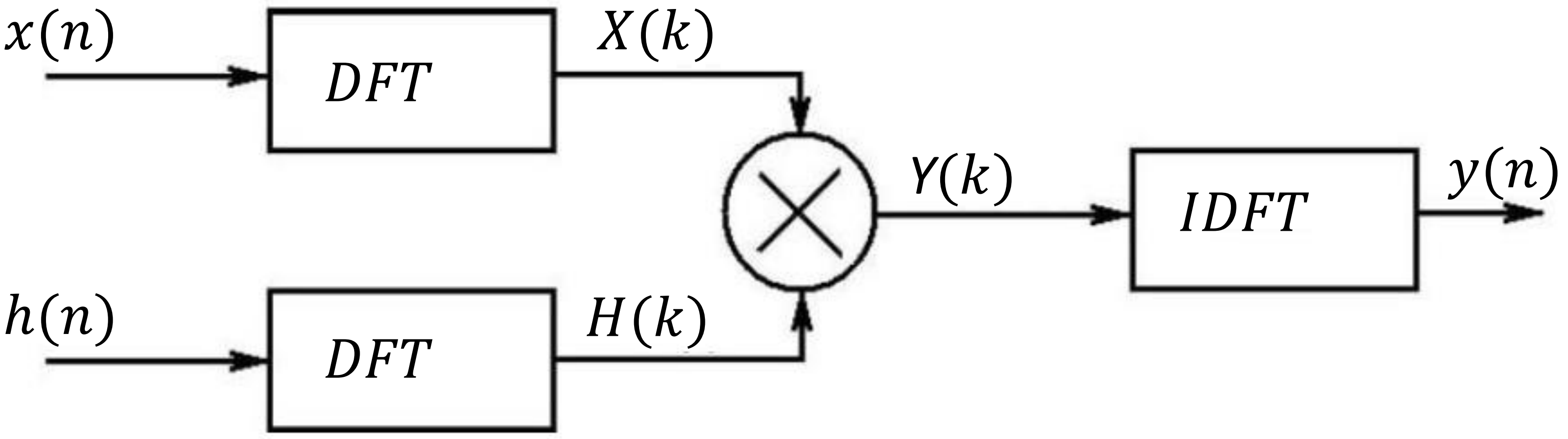


Zero-padding.

Cyclic 1D convolution

- Cyclic convolution calculation using 1D ***Discrete Fourier Transform*** (***DFT***):

$$\mathbf{y} = IDFT\left(DFT(\mathbf{x}) \otimes DFT(\mathbf{h})\right).$$



- Fast calculation of DFT, IDFT through ***FFT algorithm***.

Fast 1D Convolution Algorithms

- Convolution Algorithms
- Linear Convolutions
- Winograd Linear Convolution
- Cyclic Convolutions
- **1D FFT**
- Winograd Cyclic Convolution
- Nested convolutions
- Block convolutions
- Applications
  - Convolutional neural networks.

1D FFT

- There are various ***Fast Fourier Transform*** (***FFT***) algorithms to speed up the calculation of DFT.
- The best known is the radix-2 decimation-in-time (DIT) Fast Fourier Transform (FFT) (Cooley-Tuckey).

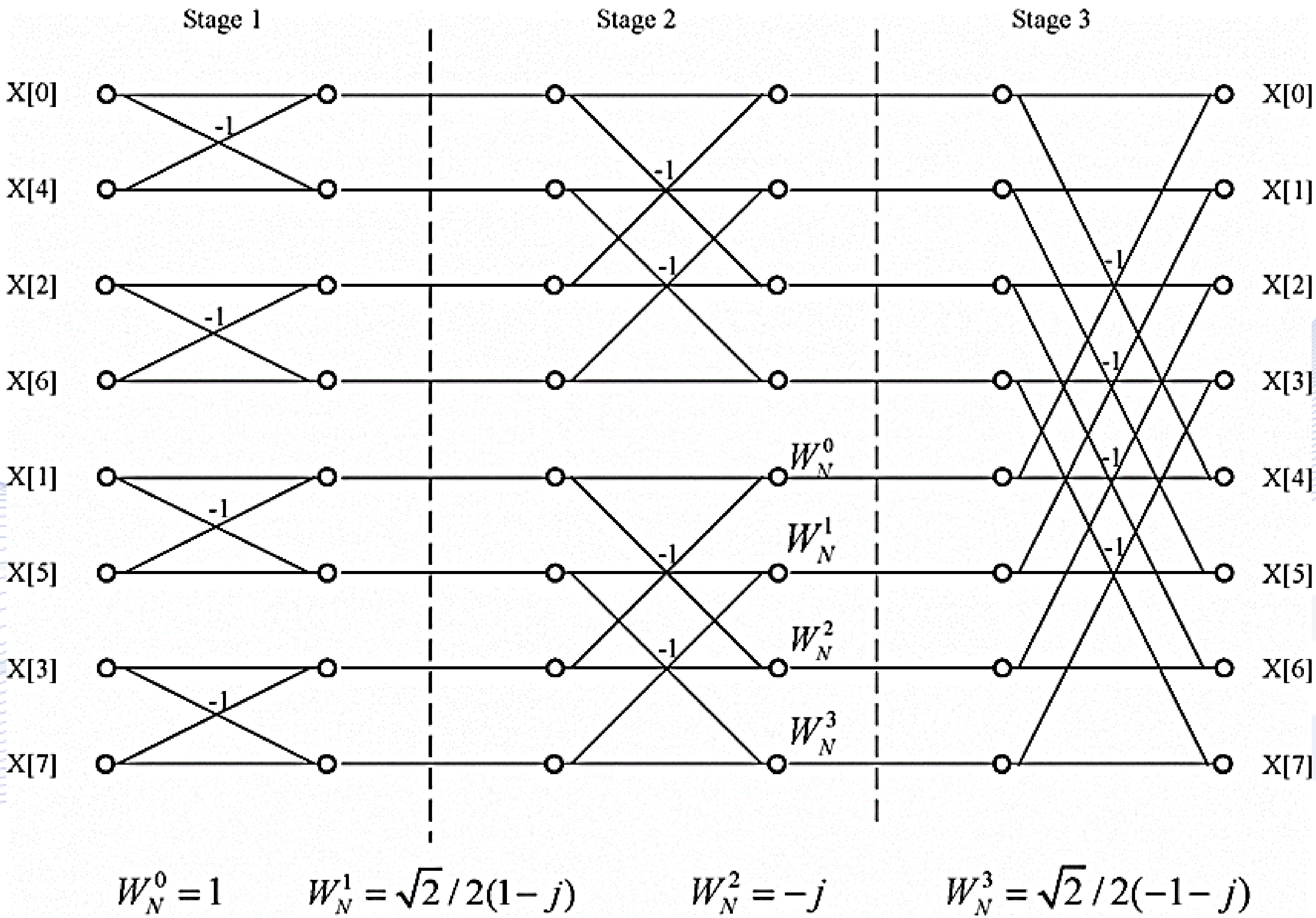
- DFT of a sequence $x(n)$of length N $(n = 0, \ldots, N-1)$:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-\frac{2\pi i}{N} nk}, \qquad k = 0, \ldots, N-1.$$

- N-th complex root of unity: $W_N{}^n = e^{-\frac{2\pi i}{N} n}, \; n = 0, \ldots, N-1.$

1D FFT

- radix-2 FFT breaks a length-N DFT into many size-2 DFTs called "butterfly" operations.
- There are $log_2 N$ FFT stages.



Fast 1D Convolution Algorithms

- Convolution Algorithms
- Linear Convolutions
- Winograd Linear Convolution
- Cyclic Convolutions
- 1D FFT
- **Winograd Cyclic Convolution**
- Nested convolutions
- Block convolutions
- Applications
  - Convolutional neural networks.

Winograd Cyclic Convolution

$\mathcal{Z}$ transform of a discrete signal $x(n)$ having domain $[0, \ldots, N-1]$ is given by:

$$X(z) = \sum_{n=0}^{N-1} x(n) z^{-n}.$$

The domain of Z transform is the complex plane, since z is a complex number.

Convolution property of the Z transform (polynomial product $X(z)H(z)$):

$$y(n) = x(n) * h(n) \Leftrightarrow Y(z) = X(z)H(z).$$

Winograd Cyclic Convolution

Polynomial product form of the 1D cyclic convolution:

$$y(k) = x(k) \circledast h(k) = \sum_{i=0}^{N-1} h(i)x\big((k-i)_N\big),$$

where: $(k)_N = k \bmod N$.

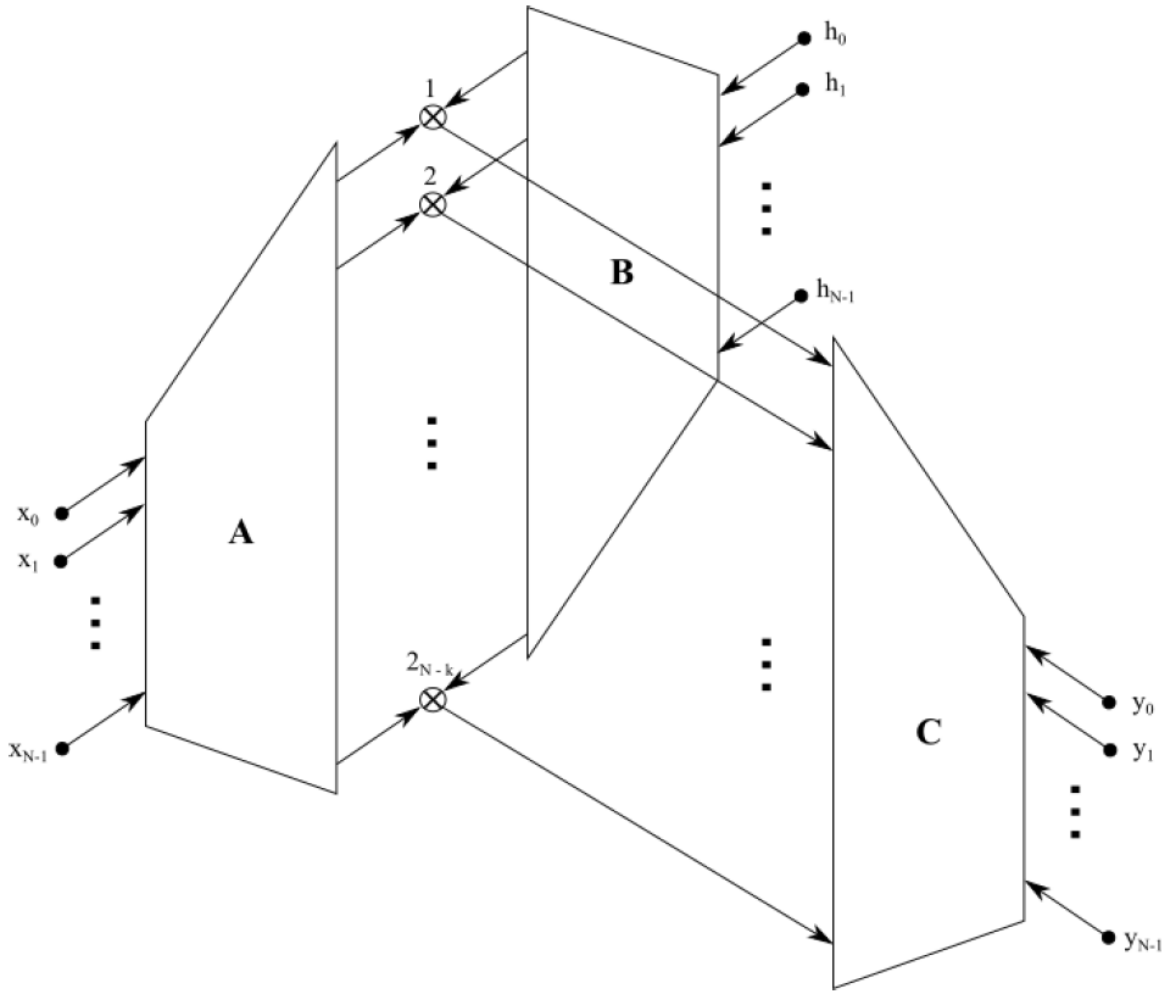
$$y(k) = x(k) \circledast h(k) <=> Y(z) = X(z)H(z) \bmod z^N - 1.$$

Winograd Cyclic Convolution

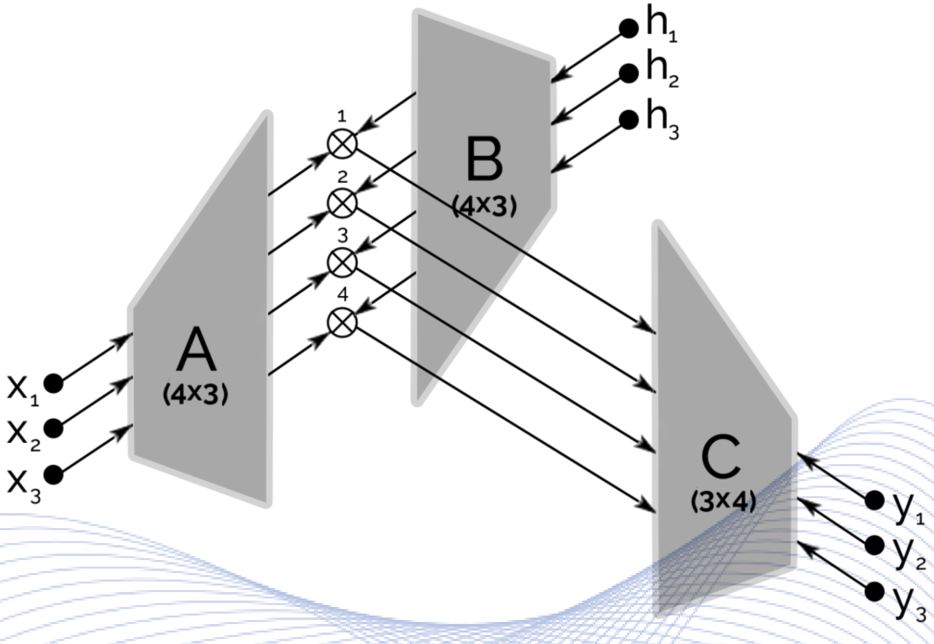
- Winograd convolution algorithms or fast filtering algorithms:

$$\mathbf{y} = \mathbf{C}(\mathbf{Ax} \otimes \mathbf{Bh}).$$

- They require only $2N - \nu$ multiplications in their middle vector product, thus having minimal complexity.
- ν: number of cyclotomic polynomial factors of polynomial $z^N - 1$ over the rational numbers Q.

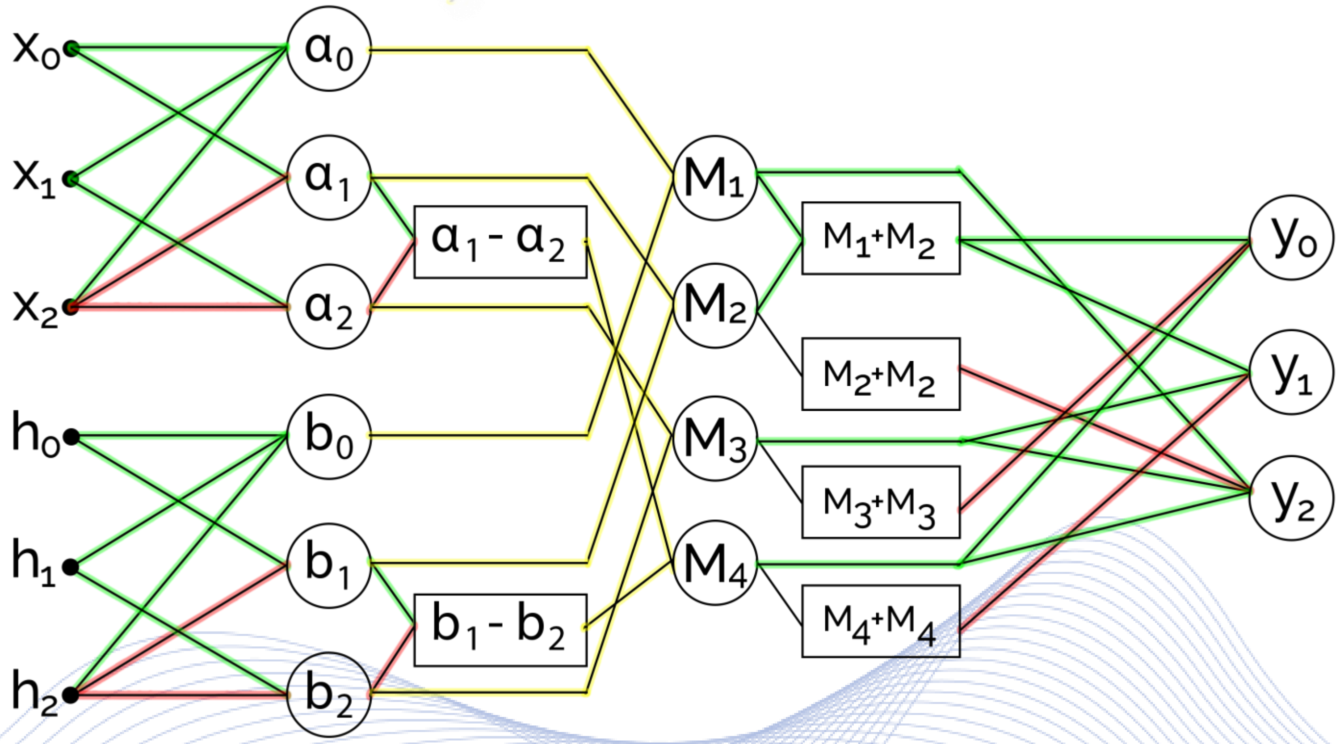


Winograd Cyclic Convolution



Block diagram of Winograd Cyclic convolution Algorithm for $N = 3$.

Winograd Cyclic Convolution



Additions	6	8	8	12	19
Multiplications	0	0	4	4	4

Block diagram of Winograd Cyclic convolution Algorithm for $N = 3$.

Winograd Cyclic Convolution

Winograd Cyclic Convolution algorithm can be equivalently expressed as:

$$\mathbf{y} = \mathbf{R}\mathbf{B}^T(\mathbf{A}\mathbf{x} \otimes \mathbf{C}^T\mathbf{R}\mathbf{h}).$$

- Matrices $\mathbf{A}, \mathbf{B}$ typically have elements $0, +1, -1$.
- Multiplications $\mathbf{C}^T\mathbf{R}\mathbf{h}, \mathbf{R}\mathbf{B}^T\mathbf{y}'$ are done only by additions/subtractions.
- $\mathbf{R}$ is an $N \times N$ permutation matrix.
- $\mathbf{C}^T\mathbf{R}\mathbf{h}$ can be precomputed.

Winograd Cyclic Convolution

- Winograd algorithm works on small blocks of the input signal.
- The input block and filter are transformed.
- The outputs of the transform are multiplied together in an element-wise fashion.
- The result is transformed back to obtain the outputs of the convolution.
- ***GEneral Matrix Multiplication*** (***GEMM***) ***BLAS*** or ***cuBLAS*** routines can be used.

Fast 1D Convolution Algorithms

- Convolution Algorithms
- Linear Convolutions
- Winograd Linear Convolution
- Cyclic Convolutions
- 1D FFT
- Winograd Cyclic Convolution
- **Nested convolutions**
- Block convolutions
- Applications
    - Convolutional neural networks.

Nested convolutions

- Winograd algorithms exist for relatively short convolution lengths, e.g.: $N = 3, 5, 7$.
- Use of efficient short-length convolution algorithms iteratively to build long convolutions.
- Does not achieve minimal multiplication complexity.
- Good balance between multiplications and additions.

Decomposition of 1D convolution into a 2D convolution:

- 1D convolution of length: $N = N_1 N_2$
- with N_1, N_2 co-prime integers, $(N_1, N_2) = 1$
- results into a 2D $N_1 \times N_2$ convolution.

Fast 1D Convolution Algorithms

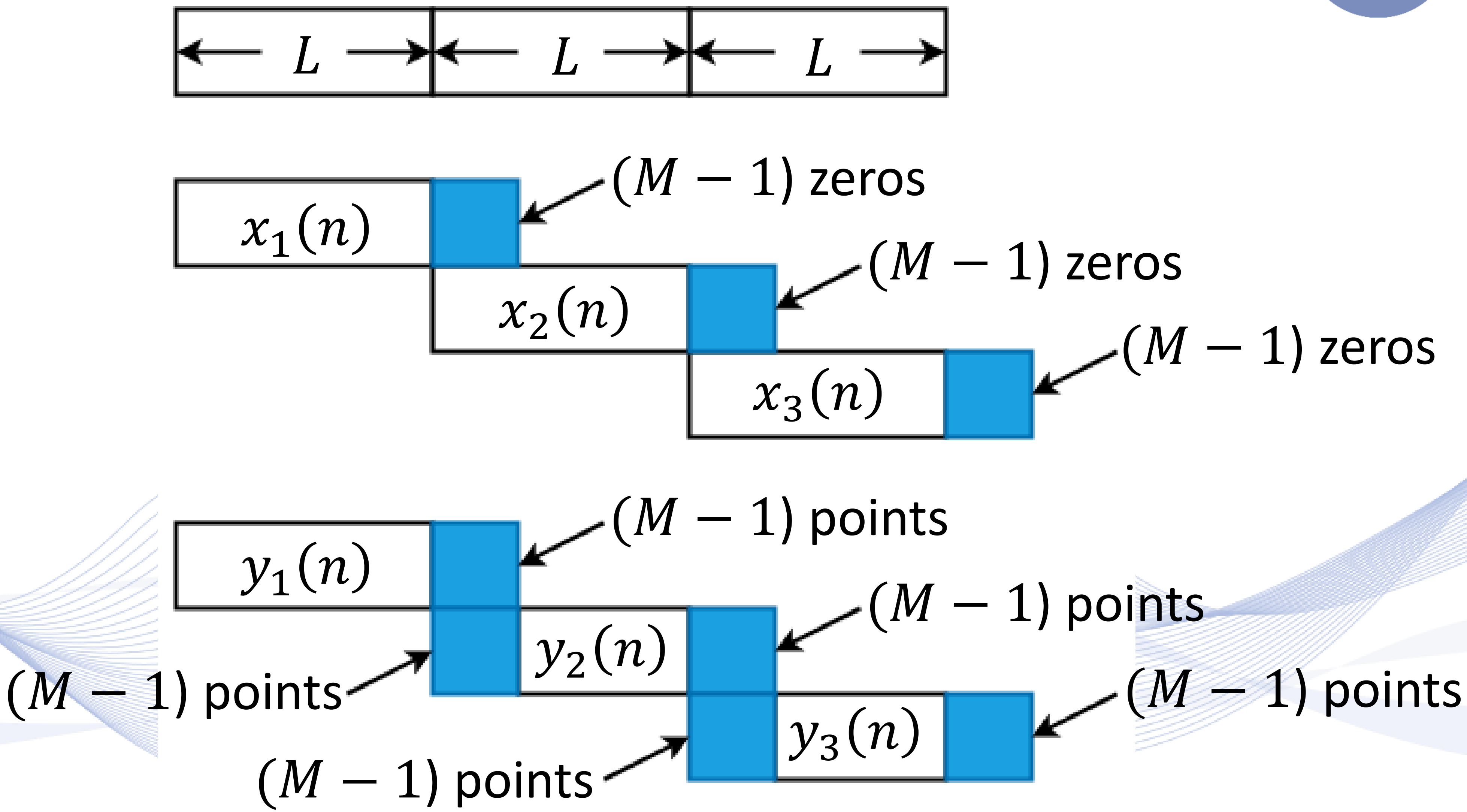
- Convolution Algorithms
- Linear Convolutions
- Winograd Linear Convolution
- Cyclic Convolutions
- 1D FFT
- Winograd Cyclic Convolution
- Nested convolutions
- **Block convolutions**
- Applications
  - Convolutional neural networks.

Block Convolutions

- Input signal $x(n)$ is split in overlapping/non-overlapping blocks.
- Blocks are convolved independently.
- Great parallelism is achieved.

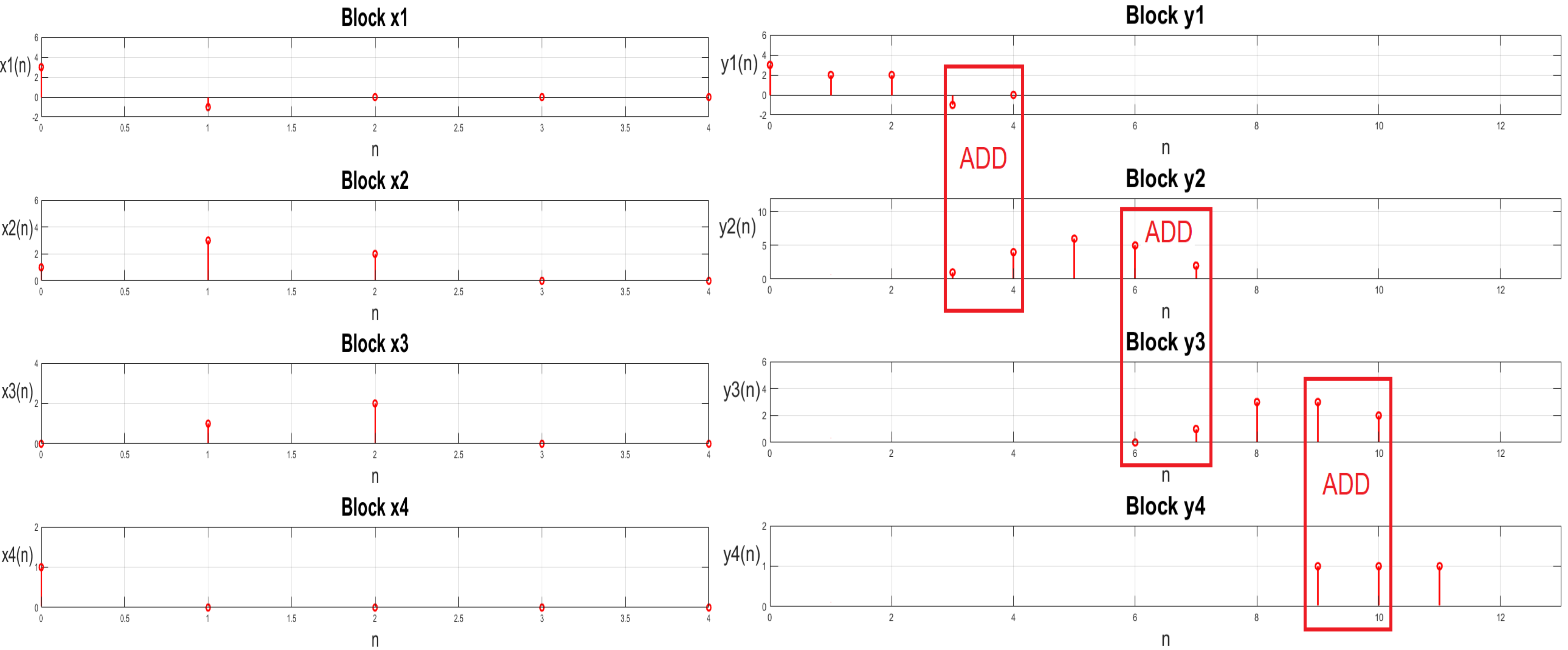
- Two block-based convolution methods:
- Overlap-add method
- Overlap-save method.

Block Convolutions



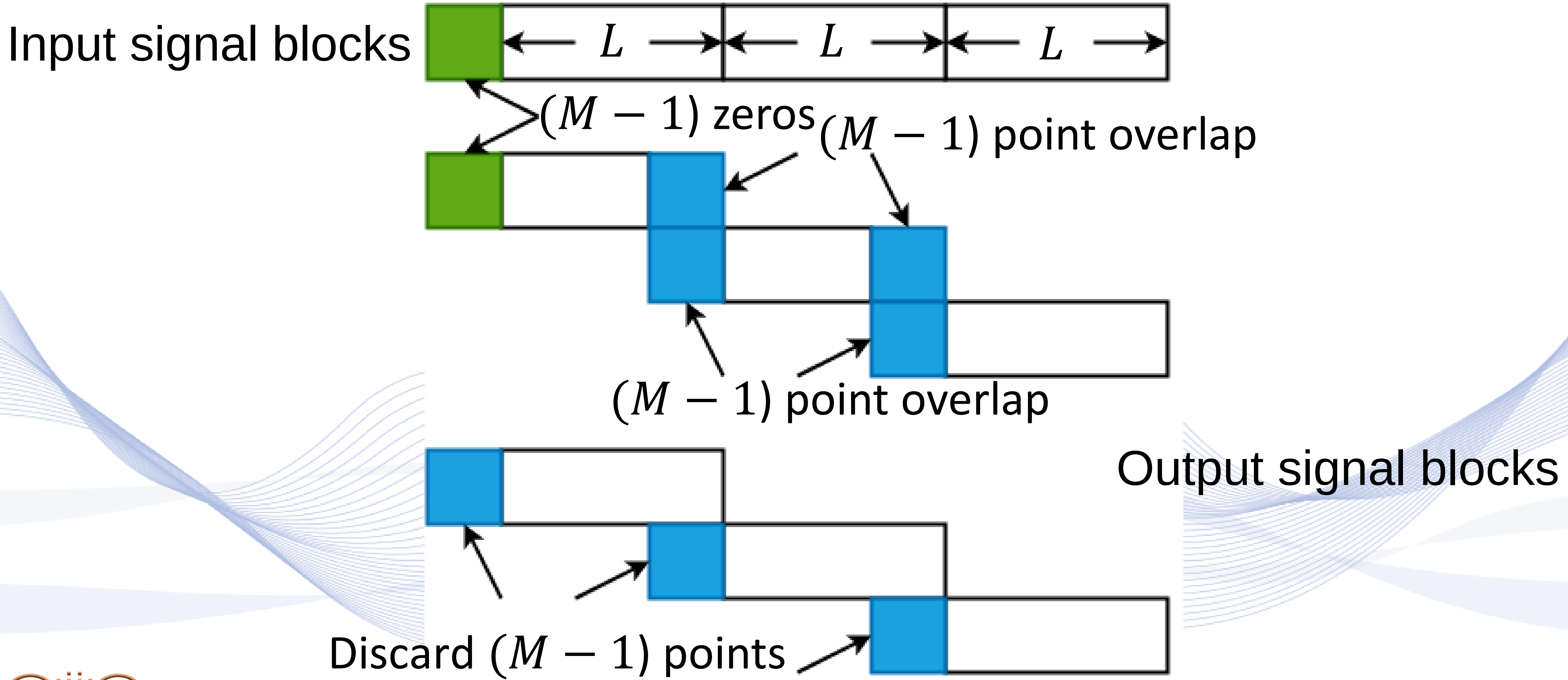
Overlap-add method.

Block Convolutions



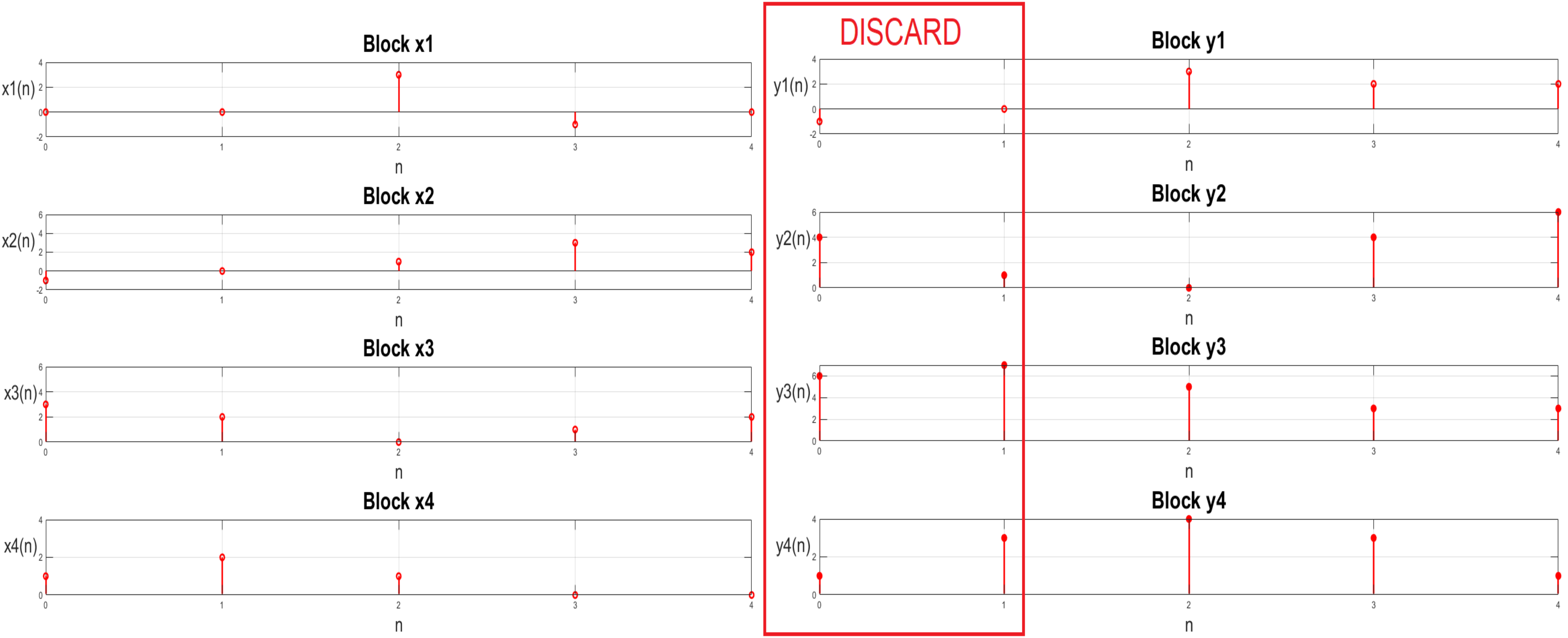
Overlap-add method.

Block Convolutions



Overlap-save method.

Block Convolutions



Overlap-save method.

Fast 1D Convolution Algorithms

- Convolution Algorithms
- Linear Convolutions
- Winograd Linear Convolution
- Cyclic Convolutions
- 1D FFT
- Winograd Cyclic Convolution
- Nested convolutions
- Block convolutions
- **Applications**
  - **Convolutional neural networks.**

Applications

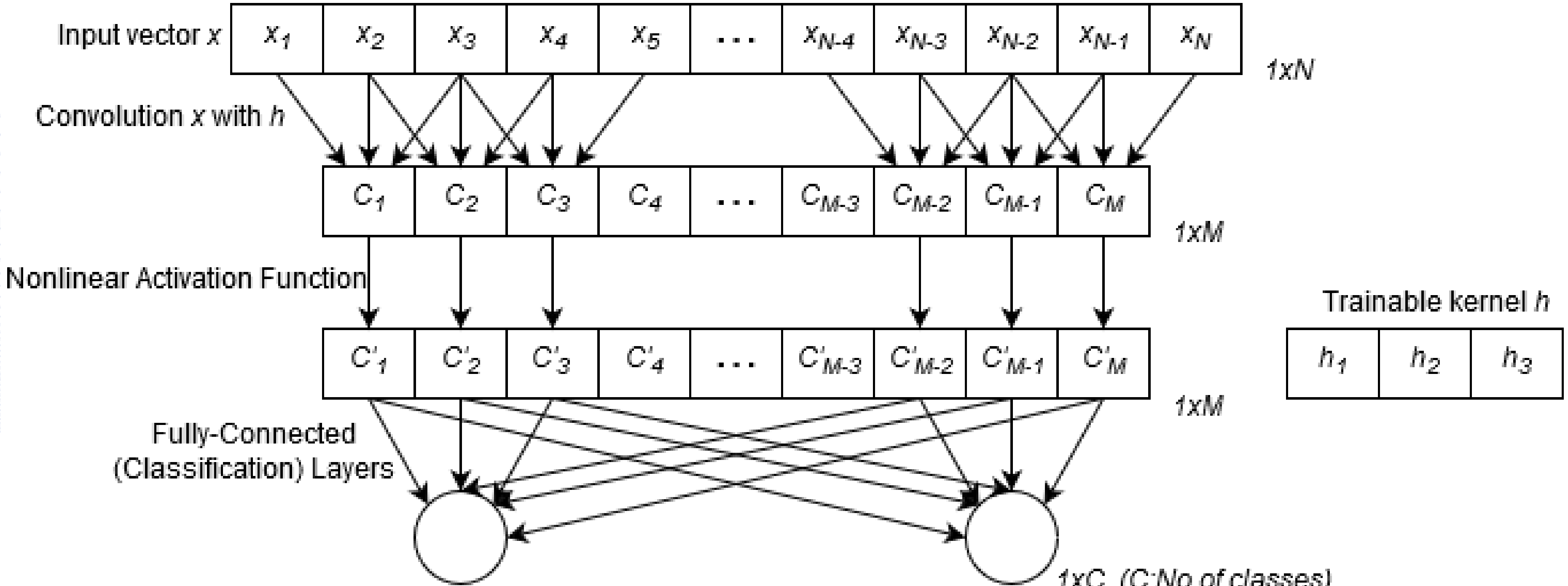
- Convolutional neural networks
- Signal processing
  - Signal filtering
  - Signal restoration
  - Signal deconvolution
- Signal analysis
  - Time delay estimation
  - Distance calculation (e.g., sonar)
  - 1D template matching

Applications

Convolutional Neural Network (***CNN***) two step architecture:

- First layers with sparse NN connections: convolutions.
- Fully connected final layers.

- Need for fast convolution calculations.



Bibliography

[OPP2013] A. Oppenheim, A. Willsky, Signals and Systems, Pearson New International, 2013.

[MIT1997] S. K. Mitra, Digital Signal Processing, McGraw-Hill, 1997.

[OPP1999] A.V. Oppenheim, Discrete-time signal processing, Pearson Education India, 1999.

[HAY2007] S. Haykin, B. Van Veen, Signals and systems, John Wiley, 2007.

[LAT2005] B. P. Lathi, Linear Systems and Signals, Oxford University Press, 2005.

[HWE2013] H. Hwei. Schaum's Outline of Signals and Systems, McGraw-Hill, 2013.

[MCC2003] J. McClellan, R. W. Schafer, and M. A. Yoder, Signal Processing, Pearson Education Prentice Hall, 2003.

Bibliography

[PHI2008] C. L. Phillips, J. M. Parr, and E. A. Riskin, Signals, Systems, and Transforms, Pearson Education, 2008.

[PRO2007] J.G. Proakis, D.G. Manolakis, Digital signal processing. PHI Publication, 2007.

[DUT2009] T. Dutoit and F. Marques, Applied Signal Processing. A MATLAB-Based Proof of Concept. New York, N.Y.: Springer, 2009

Bibliography

[PIT1987] I. Pitas and M. Strintzis, Multidimensional cyclic convolution algorithms with minimal multiplicative complexity," IEEE transactions on acoustics, speech, and signal processing, vol. 35, no. 3, pp. 384-390, 1987.

[PIT2000] I. Pitas, "Digital Image Processing Algorithms and Applications", J. Wiley, 2000.

[PIT2021] I. Pitas, "Computer vision", Createspace/Amazon, in press.

[PIT2017] I. Pitas, "Digital video processing and analysis" , China Machine Press, 2017 (in Chinese).

[PIT2013] I. Pitas, "Digital Video and Television" , Createspace/Amazon, 2013.

[NIK2000] N. Nikolaidis and I. Pitas, "3D Image Processing Algorithms", J. Wiley, 2000.

Q & A

Thank you very much for your attention!

More material in
http://icarus.csd.auth.gr/cvml-web-lecture-series/

Contact: Prof. I. Pitas
pitas@csd.auth.gr